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

Danilo Augusto Menezes Clemente

Modelagem hierárquica da disponibilidade de serviços hospedados em Centro de Dados

Recife

2022

Danilo Augusto Menezes Clemente

Modelagem hierárquica da disponibilidade de serviços hospedados em Centro de Dados

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

Área de Concentração: Redes de Computadores e Sistemas Distribuídos.

Orientador: Dr. Paulo Romero Martins Maciel

Coorientador: Dr. Jamilson Ramalho Dantas

Recife

2022

Catalogação na fonte
Bibliotecária Monick Raquel Silvestre da S. Portes, CRB4-1217

C626m Clemente, Danilo Augusto Menezes
Modelagem hierárquica da disponibilidade de serviços hospedados em centro de dados / Danilo Augusto Menezes Clemente. – 2022.
174 f.: il., fig., tab.

Orientador: Paulo Romero Martins Maciel.
Dissertação (Mestrado) – Universidade Federal de Pernambuco. CIn, Ciência da Computação, Recife, 2022.

Inclui referências.

1. Redes de computadores. 2. Modelos hierárquicos. I. Maciel, Paulo Romero Martins (orientador). II. Título.

004.6 CDD (23. ed.) UFPE - CCEN 2022-127

Danilo Augusto Menezes Clemente

“Modelagem hierárquica da disponibilidade de serviços hospedados em Centro de Dados”

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de Pernambuco, como requisito parcial para a obtenção do título de Mestre em Ciência da Computação. Área de Concentração: Redes de Computadores e Sistemas Distribuídos

Aprovado em: 20/06/2022.

BANCA EXAMINADORA

_____

Prof. Dr. Paulo Romero Martins Maciel
Centro de Informática/UFPE
(**Orientador**)

_____

Profa. Dra. Erica Teixeira Gomes de Sousa
Departamento de Computação / UFRPE

_____

Prof. Dr. Carlos Alexandre Silva de Melo
Escola de Programação Trybe

Dedico este trabalho a todos os envolvidos, direta ou indiretamente, na sua construção. A finalização deste trabalho é o resultado do esforço e contribuição de várias pessoas do âmbito pessoal e profissional. A todos, meus sinceros agradecimentos.

AGRADECIMENTOS

Primeiramente, os meus maiores agradecimentos serão sempre a Deus, à Maria Santíssima e à espiritualidade de luz por possibilitarem este momento. Sem a permissão e supervisão deles, nada poderia ter acontecido. Para eles, disponho de um imenso sentimento de gratidão.

Neste momento ímpar, não posso esquecer de todos que aceitaram e suportaram meus momentos de ausência, muitas vezes em momentos importantes, e que mesmo assim, me incentivaram constantemente. Um agradecimento muito especial aos meus pais, Carlos Clemente e Maria Felisberta, à minha esposa, Renata Cândido e aos meus, filhos Felipe e Maitê.

Gostaria de agradecer também ao Prof. Paulo Maciel, por ter me acolhido, orientado, corrigido e por toda a paciência neste período. Sem a sua determinação, este trabalho também não seria possível. Um agradecimento especial a todos os companheiros do grupo de pesquisa MoDCS, que colaboraram, incentivaram e escutaram lamentações. Em especial para Paulo Pereira e Jamilson Dantas.

Estenderei meus agradecimentos aos meus amigos e colegas de trabalho que construíram comigo a estrutura utilizada neste trabalho, discutiram melhores soluções e até suportaram períodos de ausência dos afazeres para que este trabalho fosse realizado. Sem eles, tudo seria ainda mais difícil. Enfim, gostaria de agradecer à todas as pessoas que me ajudaram direta ou indiretamente nesta conquista. Valeu a todos!

RESUMO

A computação em nuvem tem como um dos principais desafios aprimorar a utilização dos recursos físicos sem comprometer a disponibilidade dos ambientes neles hospedados. A utilização de modelos possibilita avaliação eficiente dos sistemas, permitindo melhor aproveitamento dos recursos físicos. Este trabalho se utiliza desta estratégia para conduzir avaliação de disponibilidade de um sistema de gerência acadêmica de uma grande universidade brasileira, que se encontra hospedado em uma nuvem privada. Visando obter uma visão mais genuína possível do sistema estudado, foi realizado o monitoramento das estruturas virtuais, excluindo a necessidade de realizar injeção de falhas. Os tempos de falhas e reparos, das estruturas virtuais, obtidos no estudo são resultados reais do ambiente. A infraestrutura de Tecnologia da Informação (TI) foi o foco de análise neste estudo, não considerando a infraestrutura de energia e refrigeração. Foram da infraestrutura física e lógica utilizada pelo sistema, incluindo análise de sensibilidade que identifica os componentes que possuem a maior capacidade de interferência na disponibilidade do sistema. Modelos em RBD e SPN foram concebidos, em método hierárquico, visando representar o ambiente (*baseline*), possibilitando o cálculo da disponibilidade do sistema. Também foi desenvolvida metodologia que permite a replicação adequada deste estudo, podendo ser utilizada para este sistema ou em sistemas similares. Após a correta identificação e modelagem da *baseline*, foram propostas alterações nos componentes internos ao Centro de Dados (CD), visando um aprimoramento da disponibilidade do sistema. Posterior à descoberta de um valor da disponibilidade considerado como satisfatório pelos administradores do sistema, foram adicionadas aos modelos previamente desenvolvidos, componentes físicos da estrutura do CD, em conjunto com as Causas Comuns de Falha (CCF). Neste momento do estudo, foram aplicadas técnicas de sintetização em modelos SPN, permitindo a realização dos cálculos com um baixo custo operacional. Foram concebidos modelos que possibilitaram a análise de cenários em casos de redundância de CD, permitindo a identificação do CD ativo e qual cenário resulta na melhor disponibilidade para o sistema. Como resultado deste trabalho, ao analisar e propor melhorias apenas às estruturas internas ao CD, se obteve redução de 98,5% no tempo de indisponibilidade do sistema. Quando aplicado a redundância do CD, esta redução foi de 97,36% no tempo de indisponibilidade do sistema.

Palavras-chaves: avaliação de disponibilidade; rede de petri estocástica; modelos hierárquicos; análise de sensibilidade; redundância de Centro de Dados.

ABSTRACT

Cloud computing provides an abstraction of the physical tiers, allowing a sense of infinite resources. However, the physical resources are not unlimited and need to be used more assertively. One of cloud computing's challenges is improving resource utilization without jeopardizing the availability of environments. Hierarchical models allow an efficient evaluation of the systems that use the resources of cloud computing, allowing an adequate use of the physical resources. This work proposes an availability evaluation of an academic management system of a large Brazilian university, hosted in a private cloud. To obtain a view of the system's most genuine as possible, the monitoring of virtual structures was carried out, excluding the need to perform fault injection. The failure and repair times obtained in the study are true results from the environment. The IT infrastructure was the focus of analysis in this study, not considering the power and cooling infrastructure. Preliminary studies were carried out on the physical and logical infrastructure to identify possible points for improvement in the configurations. One of them includes a sensitivity analysis that identifies the components that have the most significant capacity to interfere with the system's availability. Models in RBD and SPN were conceived, in a hierarchical method, aiming to represent the environment (*baseline*), making it possible to calculate the system's availability. An methodology was developed that allows an adequate replication of this study, which can be used in this system or similar systems. After the correct identification and modeling of the *baseline*, changes were proposed to the internal's components of CD, aiming at improving the availability of the system. After reaching an availability value considered satisfactory by the system administrators, physical components of the CD structure were added to the previously developed models, together with the CCF. At this point in the study, techniques of models' synthesis were applied, allowing the models' interaction with a low operational cost. Models were projected to permit the DC redundancy scenarios, allowing identification of the active DC and which structure allows a better system's availability. As a result of this work, a reduction of 98.5% in system downtime was obtained, when analyzing and proposing DC's internal structures improvements. When studied and applied for CD redundancy, this reduction was 97.36% in system downtime.

Keywords: availability evaluation; stochastic petri net; hierarchical models; sensitivity analysis; data Center redundancy.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

API	*Application Programming Interface*
APP	Camada de Aplicação
AS	Análise de Sensibilidade
BD	Camada do Banco de Dados
CCC	Controle do Centro de Dados
CCF	Causas Comuns de Falha
CCR	Causas Comuns de Reparo
CD	Centro de Dados
CMS	Gerenciador de Sistemas de Nuvem
COA	Capacidade Orientada a Disponibilidade
CTMC	*Continuous-Time Markov Chain*
DCS	Estrutura do Centro de Dados
DR	Recuperação de Desastre
DRaaS	*Disaster Recevery as a Service*
DRS	*Distributed Resource Scheduler*
DT	Tempo de Indisponibilidade
DTDC	Data Center Tolerante a Desastres
EFM	Modelo de Fluxo de Energia
FT	Tolerância a Falha
HA	Alta Disponibilidade
HP	Hipervisores
HTTP	*Hypertext Transfer Protocol*
HTTPS	*Hypertext Transfer Protocol Secure*
IaaS	Infraestrutura como Serviço
IC	Intevalo de Confiança

KooN	*K-out-of-N*
LAN	*Local Area Network*
LB	Camada do Balanceador de Carga
MTTF	Tempo Médio de Falha
MTTR	Tempo Médio de Reparo
NET	*Switchs Core Ethernet*
NIST	*National of Standards and Technology*
PaaS	Platforma como Serviço
PDC	Centro de Dados Principal
PN	*Petri Net*
POD	*Performance Optimization Data Center*
RBD	*Reliability Block Diagram*
SaaS	*Software* como Serviço
SAN	*Storage Area Network*
SDC	Centro de Dados Secundário
SGBD	Sistema Gerenciador de Banco de Dados
SLA	Acordo de Nível de Serviço
SO	Sistema Operacional
SPN	*Stochastic Petri Net*
SRN	*Stochastic Reward Net*
SRV	Servidores Físicos
SS	Estrutura do Sistema
STG	Unidade de Armazenamento
TI	Tecnologia da Informação
TIA	*Telecommunications Industry Association*
TTF	Tempo de Falha
TTR	Tempo de Reparo

UPS	Unidade de Alimentação Ininterrupta
VC	Camada de Orquestração
VDC	Centro de Dados Virtual
VM	Máquina Virutal
vPOD	*Virtual Performance Optimization Data Center*

SUMÁRIO

1 INTRODUÇÃO

Um desafio permanente da computação é o gerenciamento de recursos físicos dos equipamentos. Com o avanço das tecnologias, os valores dos equipamentos sofrem acréscimos constantes, o que resulta em necessidade crescente e imprescindível de utilizar ao máximo os recursos adquiridos. Esta necessidade se torna bastante aparante em ambientes que utilizam a computação em nuvem (ARMBRUST et al., 2010). Se o gerenciamento dos recursos não for eficiente, a necessidade de aquisição de equipamentos será constante, tornando os serviços ofertados cada vez mais onerosos e possivelmente inacessíveis para os clientes.

A computação em nuvem trouxe diversos benefícios aos seus usuários, sendo a melhora do desempenho, confiabilidade e disponibilidade alguns deles (AVIZIENIS A.; LAPRIE, 1986). Para atingir estes benefícios, os CDs responsáveis por hospedar a computação em nuvem, possuem diversos recursos de redundância e tolerância a falhas de equipamentos físicos e lógicos (LIU; SONG, 2003). Atualmente, existem três modelos de computação em nuvem que se destacam: nuvem privada, pública e híbrida. Cada modelo possui características particulares que se adéquam a cenários distintos, permitindo uma variedade de serviços (DILLON; WU; CHANG, 2010).

A nuvem privada é um modelo amplamente adotado em empresas que pretendem utilizar facilidades como escalabilidade, disponibilidade e desempenho em seus domínios (FURHT; ESCALANTE et al., 2010). Desta forma, essas empresas podem oferecer aos seus clientes internos uma infraestrutura ágil, flexível e um alto índice de disponibilidade, enquanto permanecem com o controle total sobre os dados (MATOS et al., 2017). Além disso, a nuvem privada pode complementar a infraestrutura local interagindo com uma nuvem pública sempre que houver necessidade de melhorar o desempenho e/ou o armazenamento.

O primeiro passo no cenário da computação em nuvem de muitas empresas que possuem um CD próprio, é a adoção da Infraestrutura como Serviço (IaaS) (ARMBRUST et al., 2010). Deste modo, essas empresas podem se beneficiar da capacidade de escalonamento e redução de custos, melhorando cada vez mais a disponibilidade dos serviços hospedados. A alocação de recursos é um dos principais desafios de uma nuvem IaaS (MANVI; SHYAM, 2014), tornando as estratégias de escalabilidade um ponto crucial do ambiente. A escalabilidade horizontal é a mais utilizada nesta categoria de nuvem por permitir melhorar a disponibilidade dos serviços, aumentando a quantidade de Máquina Virutal (VM) do ambiente (GUPTA; CHRISTIE; MANJULA,

2017). A utilização de métricas de disponibilidade e confiabilidade são amplamente utilizadas para avaliar o grau de operabilidade de um sistema ou componente (BAUER, 2011). Essas métricas combinadas com a utilização de modelagem hierárquica (BRUNEO, 2013; MACIEL; DANTAS; JÚNIOR, 2019) possuem uma alta capacidade de melhorar os custos operacionais e o planejamento da infraestrutura de TI (BAUER; ADAMS, 2012). Um dos recursos amplamente utilizados para melhorar a disponibilidade é a replicação de componentes lógicos e físicos (BAUER; ADAMS, 2012). Esta replicação podem ocorrer em vários níveis de complexidade, podendo variar entre replicações de ambientes virtuais, servidores físicos até replicações de CD (PEREIRA et al., 2020).

A redundância de CD naturalmente possibilita o aumento da disponibilidade do ambiente, porém, os benefícios são mais abrangentes. O intuito principal desta redundância é a capacidade do negócio/serviço se recuperar de forma ágil, após uma situação de desastre (NGUYEN; KIM; PARK, 2016). Esta capacidade proporciona uma redução de indisponibilidade e consequentemente redução de prejuízos financeiros para a instituição. A redundância de CD também possibilita aprimoramento no armazenamento e cópias de segurança dos dados, melhor balanceamento de carga (processamento e rede), segurança da informação, alta disponibilidade, dentre outros (BAUER; ADAMS; EUSTACE, 2011; ALSHAMMARI et al., 2017). Então, ao estruturar o ambiente possibilitando a redundância de CD, foi pensado muito além do aumento da disponibilidade dos sistemas hospedados. Neste nível de replicação, o CD pode ser classificado como ativo ou passivo. Um CD classificado como ativo recebe as requisições dos clientes e seus equipamentos estão processando e armazenando as informações geradas pelos serviços hospedados. Já o CD passivo não recebe as requisições dos clientes, porém, recebe periodicamente os resultados processados pelo CD ativo, servindo como cópia de segurança (MENDONÇA et al., 2018). Existem duas possibilidades básicas de funcionamento. A redundância chamada ativo/ativo, onde todos os CDs estão funcionais e recebendo requisições dos clientes, e a chamada ativo/passivo, onde existe um CD recebendo as requisições dos clientes e um CD em configuração de espera (BAUER; ADAMS; EUSTACE, 2011).

Os trabalhos relacionados considerados neste estudo abordam modelos que visam avaliar a disponibilidade de sistemas hospedados em nuvem, além de analisarem a disponibilidade de ambientes que se utilizam de CD redundantes. Sousa et al. (SOUSA et al., 2017) propôs modelos capazes de representar infraestruturas em nuvem com diferentes mecanismos de redundância, tais como *cold standby, hot standby, warm standby* e mecanismo de redundância ativo/ativo, além de permitir a avaliação do respectivo impacto na disponibilidade e indispo-

nibilidade.Os modelos apresentados no estudo são bem estruturados, assim como os módulos de gerenciamento da infraestrutura em nuvem, porém, as migrações das máquinas virtuais e a localização de onde estão hospedadas não foram o foco do estudo. Da mesma forma que não foi contemplado uma análise de sensibilidade para identificar componentes mais sensíveis à melhora na disponibilidade. Esses pontos foram contemplados nesta dissertação.

1.1 MOTIVAÇÃO E JUSTIFICATIVA

O índice de disponibilidade é bastante observado pelos administradores de sistemas que buscam aprimorá-lo por possuir uma relação direta com o tempo em que o sistema está acessível (KUO; ZUO, 2003; MACIEL et al., 2012a). Em sistemas hospedados em nuvem, essa busca se torna ainda mais intensa, pois a ideia oferecida pela nuvem é justamente acessibilidade e disponibilidade constante. Muitas empresas conseguem vender os seus serviços por comprovar o alto índice de disponibilidade (ARMBRUST et al., 2010). Em algumas situações como em sistemas críticos, a indisponibilidade pode causar perdas de vidas ou perdas financeiras. Independente da motivação, seja salvar vidas, angariar recursos financeiros ou preservar a qualidade de um serviço, todos almejam melhorar a disponibilidade e alcançar o patamar mínimo de cinco 9's (BAUER, 2011). Algumas estratégias podem ser utilizadas para o aprimoramento do índice de disponibilidade, tais como: replicação de componentes físicos ou lógicos e/ou replicação do CD (MESBAHI; RAHMANI; HOSSEINZADEH, 2018). Independente da estratégia utilizada, é perceptível a importância do índice de disponibilidade na representabilidade de sistemas de TI.

Este trabalho analisa um sistema responsável pela gerência acadêmica dos alunos de uma universidade brasileira de grande porte. A funcionalidade deste sistema é apoiar as áreas de ensino (graduação e pós-graduação), pesquisa, recursos humanos, processos administrativos, planejamento institucional, dentre outras funcionalidades. Este sistema possui uma estrutura bastante utilizada em computação em nuvem. Ele se utiliza de balanceadores de carga, servidores de aplicação e de banco de dados. Desta forma, este estudo pode ser aplicado em qualquer sistema que possua estruturas semelhantes ou que possam ser minimamente adaptáveis. O sistema estudado, conforme identificado através do monitoramento, possui uma disponibilidade de 99,46 %, que representa um tempo de indisponibilidade anual de $50,76\ h$. Segundo os administradores da universidade, este tempo de indisponibilidade é considerado inapropriado. De posse destas informações, foi realizada análise detalhada do ambiente focando em identificar possíveis pontos de melhorias para aprimoramento do tempo de atividade do sistema.

Como resultado do estudo, empregaram-se estratégias de aprimoramento da disponibilidade. Foram utilizadas replicações de camadas físicas e virtuais, e replicação de CD.

Neste trabalho, é apresentado o ambiente considerado ideal para o sistema estudado. Até chegar a este ambiente ideal, se apresenta uma evolução estrutural, realizando redundâncias de componentes estratégicos definidos através da aplicação da técnica de Análise de Sensibilidade (AS) (HAMBY, 1994). A evolução do ambiente é apresentada nos estudos de casos que se utilizam de técnicas de modelagens estocásticas.

1.2 OBJETIVOS

Este trabalho tem o objetivo de apresentar uma metodologia e uma estratégia de modelagem hierárquica em RBD e SPN para valiar e prover uma maior disponibilidade de serviços hospedados em centro de dados. Para atingir o objetivo principal, foi realizada análise completa da infraestrutura utilizada em um sistema de gerência acadêmica. A metodologia capacita futuras análises e continuações do presente trabalho. Foram criados também, cenários que possibilitaram o aumento da disponibilidade do sistema, utilizando-se de mecanismos de redundâncias que variam entre máquinas virtuais à redundância do CD. Esta proposta elaborou modelos hierárquicos, que auxiliam no planejamento da infraestrutura lógica e física do sistema estudado.

Por se tratar de um sistema que está sendo utilizado em um ambiente de produção, foi utilizado o monitoramento dos componentes visando o recolhimento mais fidedigno dos tempos de falhas e reparos. Identificaram-se os pontos sensíveis através da análise de sensibilidade, que permitiu assertivamente identificar componentes que possuem maior impacto na disponibilidade.

Foi realizado também um estudo sobre a estrutura do CD visando identificar os pontos mais suscetíveis a falhas e possíveis configurações de redundância do CD. Em outras palavras, os objetivos específicos são:

- Propor metodologia de avaliação para conduzir sistematicamente o estudo e possibilitar futuras replicações e/ou ampliações deste trabalho;
- Conceber modelos hierárquicos para estimar e planejar a disponibilidade do sistema estudado;

- Desenvolver modelos baseados em *Stochastic Petri Net* (SPN) para representar redundâncias (*hot e cold standby*), migrações de VMs (*live e cold migrations*, redundância do CD (ativo/passivo e ativo/ativo) e controle CD ativo;

1.3 ESTRUTURA DA DISSERTAÇÃO

O Capítulo 2 explica os principais conceitos relacionados às medidas e modelos de disponibilidade, análise de sensibilidade, configurações e redundância de CD. O Capítulo 3 resume os trabalhos relacionados encontrados na revisão da literatura. O Capítulo 4 apresenta a metodologia concebida para avaliar a disponibilidade dos serviços hospedados nos CDs. A arquitetura física e virtual do sistema estudado, com a estrutura do CD são apresentados no Capítulo 5. O Capítulo 6 apresenta os modelos concebidos. O Capítulo 7 expõe os estudos de casos deste trabalho. Neste capítulo encontram-se as análises de disponibilidade do cenário atual e cenários propostos. Na construção destes estudos, foi utilizado técnicas como AS e modelagens hierárquicas. Os modelos concebidos propõem avaliações de disponibilidade e propostas de melhorias na estrutura de ambiente, incluindo redundância do CD. Finalmente, o Capítulo 8 contém as considerações finais sobre a dissertação e discussões sobre as direções para trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo introduz os principais conceitos utilizados no estudo. Inicialmente, serão apresentados conceitos sobre estruturas de um CD, assim como visão geral sobre computação em nuvem. Em seguida, é apresentada uma introdução sobre conceitos de dependabilidade e análise se sensibilidade. Os próximos conceitos a serem apresentados são referentes às técnicas de modelagem em *Reliability Block Diagram* (RBD) e SPN.

2.1 ESTRUTURA DO CENTRO DE DADOS

Um CD é composto por conjuntos de sistemas que se interligam harmonicamente, se utilizando de diversas tecnologias que visam atender um objetivo único: fornecer condições operacionais para os sistemas hospedados. Atualmente, os CDs são projetados para atender principalmente duas categorias de serviços; computação em nuvem e *Big Data*. Estas categorias necessitam de uma estrutura cuidadosamente elaborada, projetadas seguindo metodologias de *designer*. Este assunto é amplamente estudado em (BONNEVILLE; PEKELNICKY, 2014; BARROSO; CLIDARAS; HÖLZLE, 2013; GREENBERG et al., 2009; RAHMAN; ESMAILPOUR, 2016; HEADQUARTERS, 2007; BARROSO; HÖLZLE; RANGANATHAN, 2018).

A estrutura de um CD possui três conjuntos de equipamentos: sistema de fornecimento energético, sistema de refrigeração e equipamento de TI (RAHMAN; LIU; KONG, 2013). O sistema de fornecimento energético é composto por unidades de conversão de energia (correntes contínuas para alternadas e alternadas para contínuas), reguladores de tensão e Unidade de Alimentação Ininterrupta (UPS). A UPS possui a função de evitar a interrupção energética aos equipamentos do CD. O sistema de refrigeração possui os equipamentos de ar-condicionados, podendo ser utilizados sistemas centrais ou equipamentos individualizados. Por último, são os equipamentos de informática propriamente ditos, compostos por: servidores, unidades de armazenamento e equipamentos de rede para comunicação de dados.

Os estudos (TIA, 2021; GENG, 2014) recomendam que um CD possua um conjunto de características, dentre elas:

- Resiliência: capacidade de se recuperar rapidamente de uma falha de equipamento ou desastre natural;
- Confiabilidade: capacidade do equipamento desempenhar satisfatoriamente uma deter-

minada função;

- Disponibilidade: capacidade de um componente estar em um estado funcional para executar uma função exigida;

Existem alguns guias internacionais que padronizam a estrutura do CD. Esta padronização é importante para identificar a capacidade operacional e a disponibilidade de cada CD (BARROSO; HÖLZLE; RANGANATHAN, 2018; GENG, 2014). Estes guias contemplam padronizações para redundâncias de telecomunicações, estruturas física e elétrica. O guia utilizado neste estudo é o **ANSI/TIA-942-A**, desenvolvido pela *Telecommunications Infrastructure Standard for Data Center* (TIA, 2021). Este guia cria definições referentes à redundância dos componentes de um CD. Segundo o guia, a terminologia **redundância N** é utilizada para representar componentes únicos, que não possuem redundância nas suas configurações, ou seja, que possuem o requisito mínimo para o funcionamento do serviço ofertado.

A definição **redundância N + 1** é utilizada para componentes do CD que possuem um equipamento adicional ao requisito mínimo. Este equipamento adicional pode ser um módulo de um equipamento específico, um servidor ou *switch* adicional, podendo ser até um caminho de rede *ethernet* ou rede elétrica que forneça alguma redundância. Um exemplo é a utilização de UPS ou gerador para contemplar o fornecimento elétrico do CD, pois além do fornecimento elétrico oriundo da estação de energia, possuirá uma capacidade de fornecimento energético extra.

A terceira definição apresentada no guia é a **redundância N + 2**, que representa os componentes que possuem duas unidades, módulos, caminhos ou sistemas adicionais além do requisito mínimo. Um exemplo é o CD conseguir ser compensado por uma interrupção energética utilizando um UPS e gerador. Neste exemplo, existem duas redundâncias energéticas. Este mesmo pensamento pode ser estendido a equipamentos, conexões de rede, sistema de refrigeração, sistema de prevenção de incêndio, etc.

Existe também a **redundância 2N**, utilizada para componentes que possuam o dobro da capacidade mínima para operar. Um exemplo é a utilização de duas empresas de fornecimento de energia. No entanto, essas empresas devem ter diferentes subestações elétricas e a entrada elétrica no CD devem ser por caminhos distintos. Outro exemplo é a utilização de dois *switchs* operando em redundância de forma ativo/ativo. Todos os equipamentos conectados em um *switch*, precisam estar conectados no segundo *switch* como redundância.

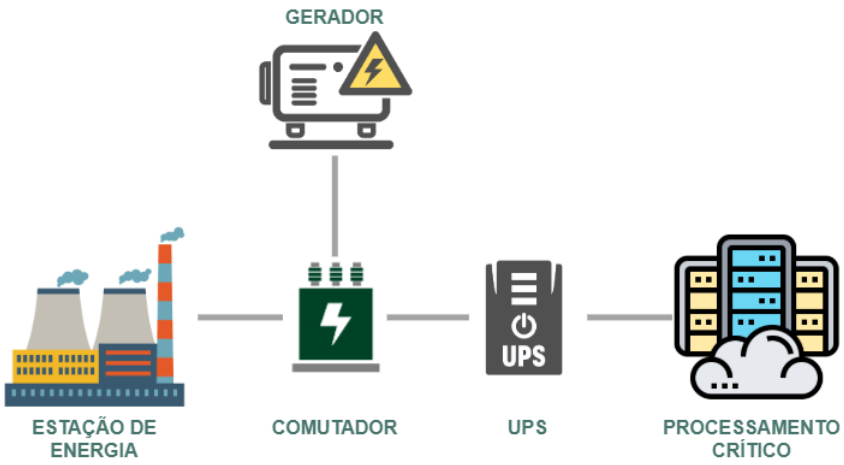
A última definição apresentada é a **redundância 2(N + 1)**, que classifica componentes que possuam duas unidades completas (N + 1), módulos, caminhos ou sistemas. Existe uma redundância para cada dispositivo. Utilizando o exemplo anterior, o CD precisa ter um UPS ou gerador para cada empresa de fornecimento energético.

A *Uptime Institute*[1], uma organização de serviços profissionais especializada em CDs, e a *Telecommunications Industry Association* (TIA), defendem uma classificação baseada na redundância dos sistemas de fornecimento energético e refrigeração, (IV et al., 2006; FERREIRA et al., 2018; BARROSO; HÖLZLE; RANGANATHAN, 2018; GENG, 2014; VERAS, 2009). A nomenclatura adotada é o *tier*, que varia entre *tier I* à *tier IV*.

A primeira classificação é o ***Tier* I**, também conhecido como **sistema básico**. Os CDs com esta classificação possuem apenas uma estrutura para fornecimento energético e contém um ou vários equipamentos de refrigeração, porém sem redundância. Esta classificação é utilizada para CDs que não contemplam componentes redundantes (nem físicos, nem lógicos), possuindo similaridade com o nível de redundância (N). Esta classe fornece um nível mínimo de distribuição de carga com pouca ou nenhuma redundância. Um defeito ou manutenção programada pode levar à interrupção dos serviços hospedados. Os CDs com esta classe apresentam pontos de falha na sua estrutura que geram indisponibilidade, como: defeito no fornecimento energético; defeito no equipamento de telecomunicações; e defeito em roteadores e switches quando não redundantes;

A Figura 1 representa um CD Tier I. Em um CD com esta classe não há componentes redundantes para o sistema de energia (estação de energia, UPS e gerador).

Figura 1 – Centro de Dados *Tier* I: subsistema energético.



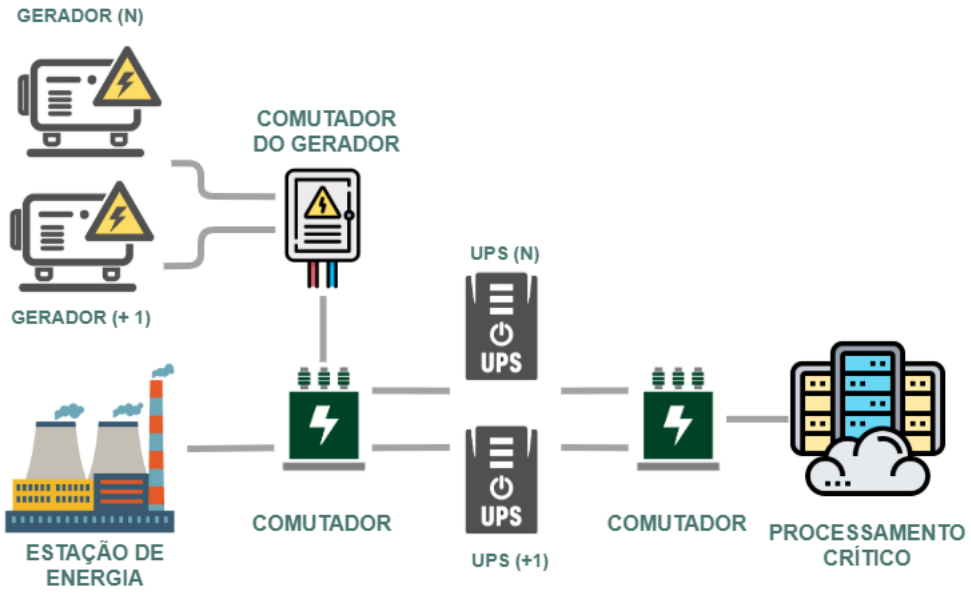
Baseada em: FERREIRA et al. (2018)

[1] https://uptimeinstitute.com

A segunda classificação é o ***Tier* II**, identificado como **componentes redundantes**. Esta classe é utilizada para CDs que possuem alguns componentes redundantes e equivale à combinação de (N + 1) e (N + 2). Nestes CDs existem redundâncias parciais em energia, resfriamento e rede (*Local Area Network* (LAN) e *Storage Area Network* (SAN)). Os recursos necessários para que o CD se enquadre nesta classe são: existência de switches LAN e SAN com módulos redundantes; cabeamento redundante para o *backbone* principal (LAN e SAN); redundância para equipamentos do sistema energético (UPS e gerador); e existência de um sistema de refrigeração operando ininterruptamente, 24x7 - 365 dias, possuindo redundância mínima N + 1.

Os CDs com esta classe apresentam pontos de falha relacionados ao sistema de refrigeração e energia. A Figura 2, expõe um CD Tier II. Na Figura percebe-se a existência de redundância no fornecimento de energia (N + 1) originada pelo gerador e UPS.

Figura 2 – Centro de Dados Tier II: subsistema energético.



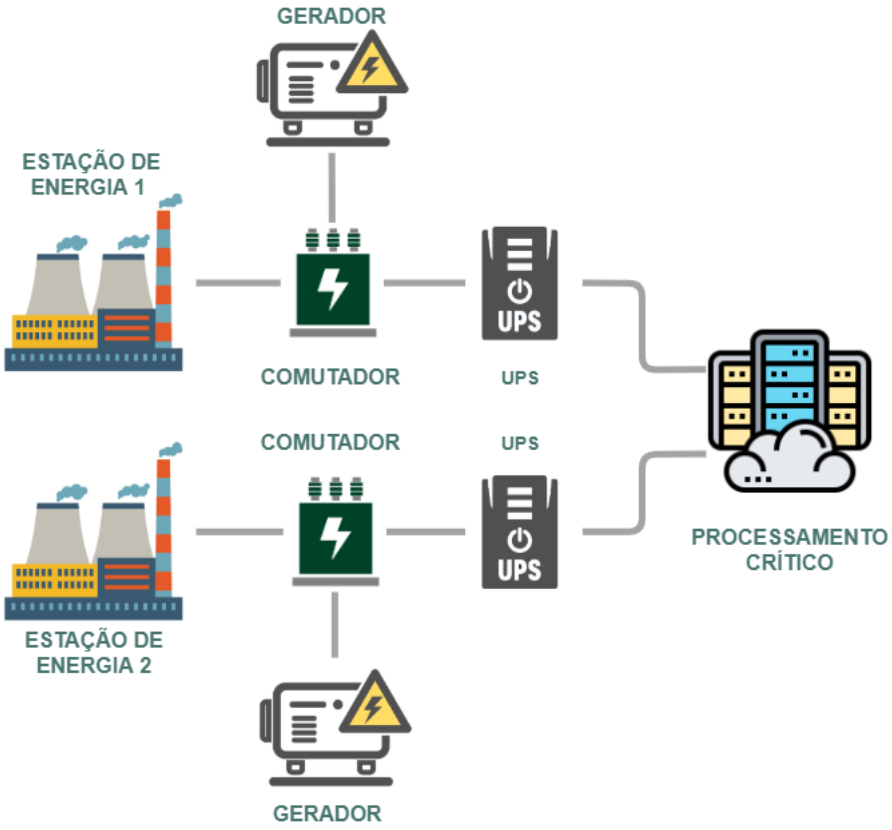
Baseada em: FERREIRA et al. (2018)

Já a terceira classe é o ***Tier* III**, identificada como **manutenção simultânea**. Os CDs contemplados com esta classificação são conhecidos por possuírem um sistema autossustentável. Estes CDs possuem dois caminhos energéticos completos, que podem estar operacionais simultaneamente, possibilitando manutenções corretivas ou preventivas sem interrupções dos serviços ofertados. Isto significa que todo equipamento pode ser removido/substituído de forma planejada, sem interromper os recursos de TI para o usuário final. As características desta classe equivale à definição de redundância (2N) sendo utilizadas por grandes empresas.

Este *tier* apresenta os seguintes recursos: duas empresas de telecomunicações utilizando conexões distintas e distância mínima de 20 metros entre elas; redundância de pelo menos (N + 1) para o sistema energético; e a quantidade de geradores precisa ser de (N + 1) e possuir uma autonomia energética de pelo menos 72 horas em momentos de interrupção energética.

O único ponto de falha para estes CDs é a sala de distribuição, onde os switches Core, LAN e SAN estão instalados. A Figura 3 expõe uma CD Tier III. Nesta estrutura de CD, há redundância de todos os componentes do sistema de distribuição de energia (estação de energia, gerador e UPS).

Figura 3 – Centro de Dados Tier III: subsistema energético.



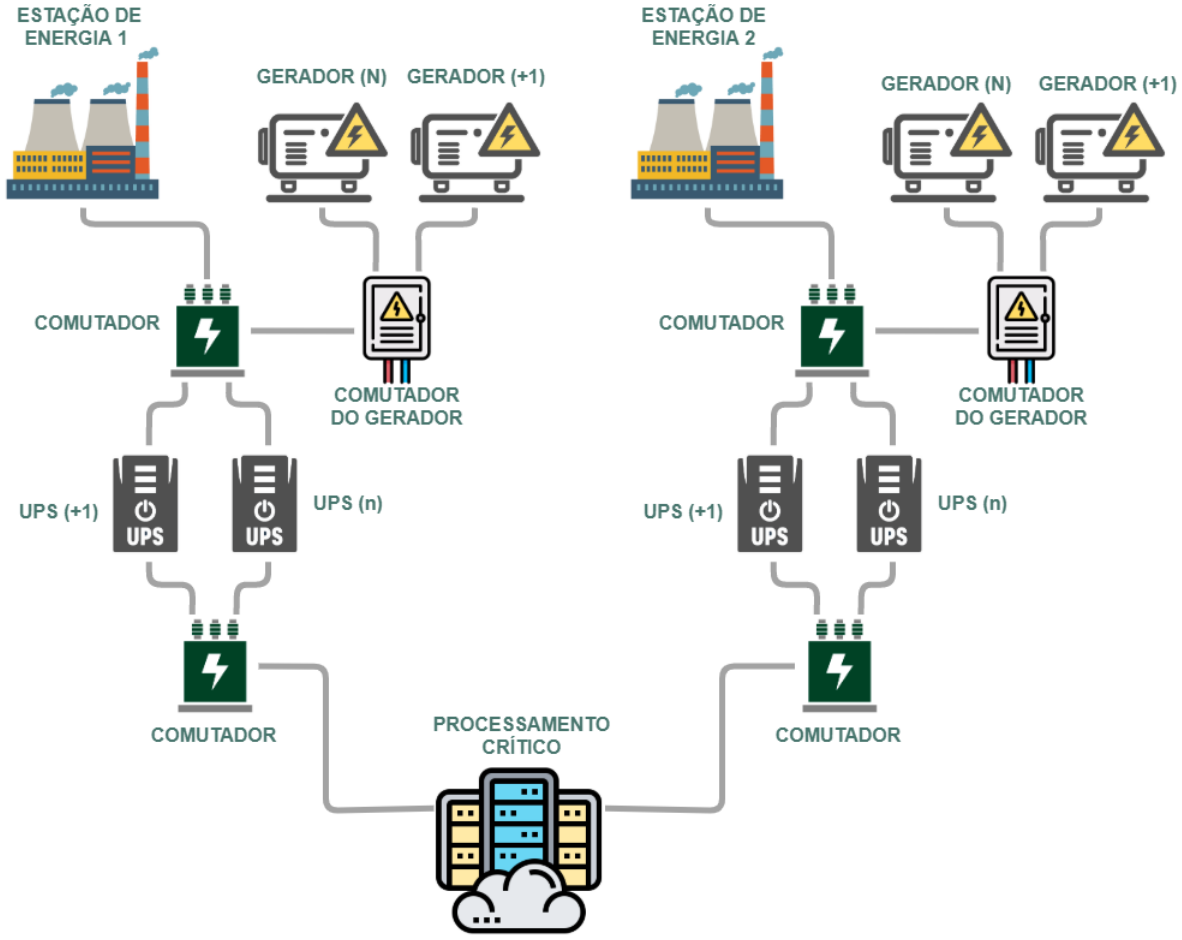
Baseada em: FERREIRA et al. (2018)

A última classe é o ***Tier* IV**, **tolerante a falha**. Esta classe é conhecida como alta tolerância a falhas ou totalmente redundante. Os CDs que recebem esta classificação possuem dois caminhos de distribuição de energia e resfriamento ativos simultaneamente com componentes redundantes, possibilitando uma redundância completa ao ambiente. Este CD permite manutenção simultânea, mesmo quando há um defeito em qualquer lugar da instalação, sem causar indisponibilidade dos serviços ofertados. Supõe-se que o CD tolere qualquer defeito de equipamento sem impactar na disponibilidade. Esta classe equivalente à definição de redun-

dância 2(N + 1) e possuem: todo o *backbone* redundante, além de ser protegido por dutos fechados; equipamentos de rede redundantes (roteadores, modens de operadoras e switches LAN e SAN); infraestrutura totalmente redundante, 2(N + 1). A principal diferença entre os CDs com classe *tier III* e *Tier IV* é que o *Tier IV* utiliza duas empresas de fornecimento de energia. Ainda assim, eles devem possuir diferentes subestações elétricas e entradas elétricas distintas no CD; e estrutura de geradores que possibilitem uma autonomia energética de 96 horas.

A Figura 4 expõe um CD Nível IV.

Figura 4 – Centro de Dados *Tier* IV: subsistema energético.



Baseada em: FERREIRA et al. (2018)

A Tabela 1 apresenta uma comparação entre os CDs com as classificações apresentadas (*Tier I a IV*). Nesta tabela pode-se verificar a métrica de disponibilidade, quantidade de número de noves (#9's) e *downtime* anual em horas de cada CD com as devidas classes. Os valores foram obtidos através de (UPTIME, 2021; COLOCATION, 2021).

Tabela 1 – Relação entre Tier e disponibilidade.

Tier	Disponibilidade	#9's	dt(h)
I	0,99671	2,4828	28,82
II	0,99749	2,6003	21,99
III	0,99982	3,7447	1,58
IV	0,99995	4,3010	0,44

Fonte: COLOCATION, 2021

A complexidade da construção e estruturação de um CD foi exposta anteriormente. Foi visto também a relação da disponibilidade com a estrutura de um CD. No entanto, em muitos casos não são possíveis realizar melhorias na estrutura de um CD. Um exemplo é a não existência de duas ou mais empresa fornecendo estações de energia. Dessa forma, o CD sempre terá esse ponto de falha e a disponibilidade dos sistemas hospedados se torna limitada. Assim, é necessário utilizar outras estratégias como redundância CD, também conhecida como sites distribuídos. O intuito principal da redundância do CD é a capacidade do negócio/serviço se recuperar de forma ágil, após uma situação de desastre, proporcionando uma redução da indisponibilidade dos serviços hospedados e consequentemente de prejuízos financeiros para a instituição. Todavia, os seus benefícios não se resumem unicamente a esta capacidade. A redundância de CD possibilita um aprimoramento no armazenamento e cópias de segurança dos dados, um melhor balanceamento de carga (processamento e rede), segurança da informação, alta disponibilidade, dentre outros.

A expressão (N + K) pode definir redundância de CD, onde N representa o número de sites necessários para atender a carga projetada e K a quantidade de sites redundantes (BAUER; ADAMS, 2012). Todos os CD classificados por N estão ativos e recebendo tráfego de acesso. A redundância do CD pode seguir duas estratégias, denominadas *standby* e balanceamento de carga. Na redundância *standby*, K é configurado como *backup* de todos os sites ativos. Caso qualquer site ativo falhe, o CD de *backup* é ativado para substituir os serviços. Na configuração de balanceamento de carga, todos os CD estão ativos e compartilhando toda a carga de processamento. Se algum site falhar, o processamento será dividido entre os sites restantes.

Na elaboração de um projeto para um CD redundante, precisam ser analisadas perspectivas significativas como localização física, análises climáticas e até estabilidade política e econômica (GENG, 2014; KLEYMAN, 2013). Aspectos como capacidade de redundância do fornecimento

elétrico, riscos naturais (por exemplo, terremoto, tsunami, furacão, tornado e vulcão), topografia, infraestrutura de fibra óptica com múltiplas conectividades e até incentivos fiscais municipais e estaduais precisam ser analisados. Existem também os lugares considerados inadequados para construção, como: próximo à rios, lagos, oceanos e fundos de vales, por existir risco de inundações, tsunamis, etc.; próximo de aeroporto, pois existe risco potencial de acidente; localização com histórico ou sujeitos a terremotos e/ou tornados; países ou lugares com conflitos civis; e localização com risco de deslizamentos e risco de incêndio;

Por outro lado, existem localizações identificadas como apropriadas para a construção, como: lugares que possuam proximidades com acessos às estradas principais; estejam próximo de fornecedores energéticos; que sejam próximo de centros de serviço; e em condomínios comerciais específicos para Centro de Dados.

2.2 COMPUTAÇÃO EM NUVEM: UMA VISÃO GERAL

A computação em nuvem é um paradigma que trouxe para seus utilizadores uma abstração geográfica, possibilitando a utilização de serviços de informática sem a preocupação com a estrutura física de onde estão hospedados. Este paradigma utiliza de diversas tecnologias que possuem conceitos de virtualização (MELL; GRANCE, 2011b). A ideia principal da computação em nuvem é a oferta dos serviços sob demanda, permitindo que os clientes paguem apenas o que foi devidamente consumido (ARMBRUST et al., 2010; HAYES, 2008). Atualmente a computação em nuvem possibilita oferta de itens como serviços, energia, rede, processamento e armazenamento. A localização física das informações e recursos não é mais fundamental. O mais importante é que os recursos estejam acessíveis de forma confiável no momento desejado. Para atender às demandas de muitos usuários é necessário compartilhar recursos de computação, permitindo provisionamento rápido e escalonado (DILLON; WU; CHANG, 2010), introduzindo o conceito de elasticidade. Os benefícios da computação em nuvem aproximam cada vez mais as empresas que não pretendem ou não conseguem desprender de recursos financeiros para implementar e manter um CD (ZHANG; CHENG; BOUTABA, 2010).

Segundo (MELL; GRANCE, 2011a), publicado no *National of Standards and Technology* (NIST), o modelo de nuvem é composto por cinco características essenciais, quatro modelos de implantação e três modelos de serviço.

As características essenciais são: o **autoatendimento sob demanda**, pois permite que os usuários possam administrar os recursos contratados sem interação com o provedor do serviço.

A gerência dos recursos é realizada por ferramentas previamente configuradas; o **amplo acesso à rede**, por possibilitar que os recursos contratados estejam facilmente alcançáveis na nuvem, podendo ser acessados por qualquer meio que possua acesso à internet, como telefones, *tablets*, computadores; o ***pool* de recursos**, que permite que os recursos contratados como servidores, armazenamento, memória e largura de banda sejam configurados com uma abstração de alto nível. Desta forma, os clientes não possuem o controle da sua localização exata; a **elasticidade**, por possibilitar a alocação dinâmica de recursos físicos conforme a demanda sazonal de cada serviço. Esta característica permite um melhor aproveitamento dos recursos físicos e uma exata identificação da quantidade de recursos utilizada pelos clientes. Ao consumidor, os recursos disponíveis para provisionamento muitas vezes parecem ser ilimitados e podem ser reservados em qualquer quantidade a qualquer momento; e o **serviço medido**, pois fornece controle sobre os recursos, possibilitando o rastreamento do consumo. Isto também permite que os provedores de nuvem ofereçam um modelo pré-pago.

Com a utilização crescente da computação em nuvem, várias empresas iniciaram sua jornada de migração para esta plataforma, porém, cada uma possui características e necessidades distintas. Atualmente, a computação em nuvem pode ser segmentada em quatro conjuntos de modelo de implantação (MELL; GRANCE, 2011a).

A **nuvem privada** é fornecida por uma instituição para uso interno. A instituição se utiliza da infraestrutura de nuvem, porém, nos seus domínios, e, com o intuito de fornecer os serviços para os clientes internos, possibilitando melhor controle e privacidade dos dados. **A nuvem pública** é uma infraestrutura fornecida para uso aberto ao público. Este modelo de implantação pode ser pertencente, administrado e operado por uma organização empresarial, acadêmica ou governamental. Este modelo está presente nos provedores de nuvem. Por último a **nuvem híbrida** é uma composição de duas ou mais nuvens distintas (privadas ou públicas). A principal característica deste modelo de implantação é a possibilidade de balancear a carga entre as nuvens. Os serviços podem funcionar em várias nuvens simultaneamente, ou, variar entre nuvens em momentos específicos.

A Figura 5 apresenta visualmente as conexões entre as nuvens pública, privada e híbrida.

Figura 5 – Modelos de computação em nuvem.

Fonte: SCURRA (2022)

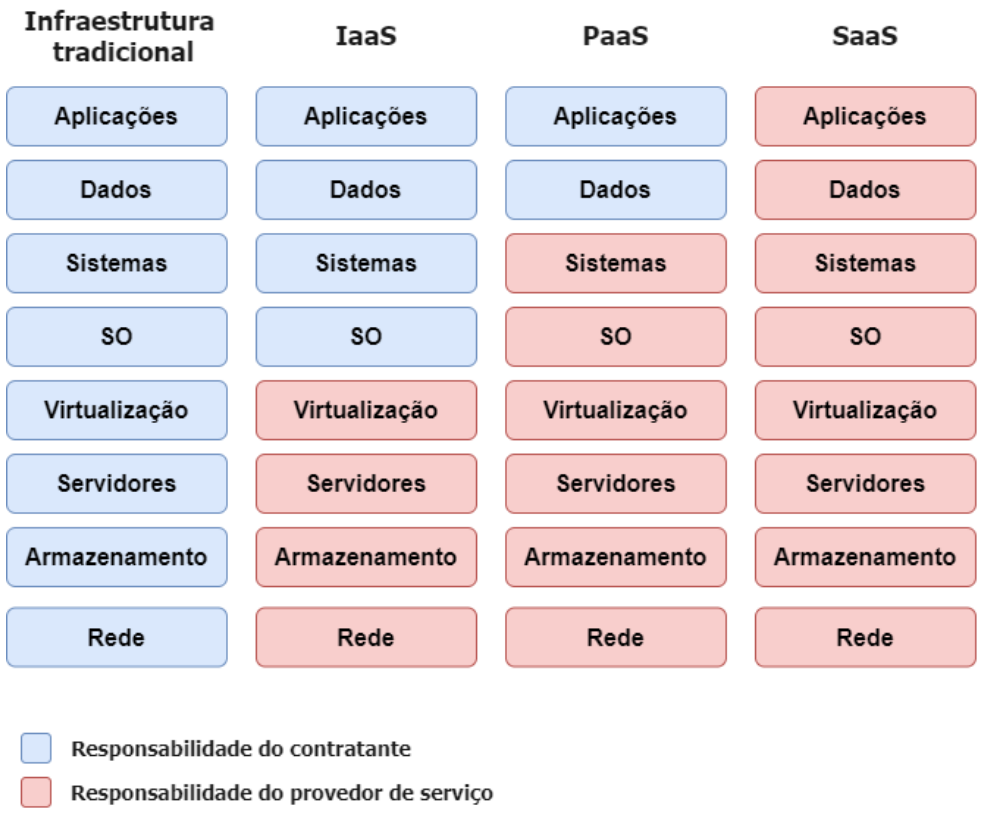
O modelo do negócio dos serviços ofertados pela computação em nuvem também sofreram alterações e adaptações ao longo do tempo. Atualmente, os serviços variam entre ofertas de *hardware* e *software* sob demanda. Estes serviços são divididos em três categorias: IaaS, Platforma como Serviço (PaaS) e *Software* como Serviço (SaaS) (GONG et al., 2010; REDHAT, 2022).

O modelo **IaaS** é o que mais se assemelha a um CD tradicional, porém, as demandas de gerência do ambiente são realizadas pelo prestador de serviço em nuvem. Assim, a gerência dos servidores, redes e unidades de armazenamento são de responsabilidade dos provedores. O cliente contrata um servidor com características específicas. Nele, pode-se instalar o Sistema Operacional (SO) e os serviços desejados. Toda a gerência do SO, serviços instalados e *backup* é de responsabilidade do cliente. Já no modelo **PaaS**, o cliente não gerencia o SO nem o serviço. O foco do cliente é utilizar um serviço ou sistema previamente instalado e configurado. Um exemplo é a contratação de uma instância de banco de dados, onde o cliente não possui a responsabilidade de instalar, configurar e realizar o *backup* do banco da dados, apenas se responsabiliza pelo serviço já instalado. Esta configuração é bastante utilizada para: desenvolvimento, testes, compilações, dentre outras. O terceiro modelo é o **SaaS**, que está voltado para o usuário final. Nesta configuração se oferta um conjunto de *softwares* como um serviço, que visa substituir os *softwares* que poderiam estar instalados no computador do cliente. Ao invés de adquirir um *software* específico para ser instalado no seu computador, o cliente pode contratar um serviço do mesmo *software*, ou similar, que será acessível a partir

de qualquer lugar, estando disponível todo o tempo.

A Figura 6 expõe um diagrama de comparação entre os modelos de serviços utilizados em nuvem e um modelo tradicional de CD. Nesta Figura pode-se perceber de quem é a responsabilidade de cada atividade. Os quadros destacados em azul são de responsabilidade do contratante, já os em vermelho são de responsabilidade do provedor de serviços.

Figura 6 – Diagrama comparativo dos modelos de serviços em nuvem.



Baseada em: REDHAT (2022)

2.3 DEPENDABILIDADE

A dependabilidade de um sistema é a capacidade do sistema prover serviços cujos resultados sejam verossímeis. (AVIZIENIS; LAPRIE; RANDELL, 2001; LAPRIE, 1985). Nesta área de estudo estão introduzidas medidas como a disponibilidade, confiabilidade e segurança.

2.3.1 Disponibilidade

A disponibilidade como medida de avaliação da operabilidade em serviços de computação vem sendo estudada há muito tempo (AVIZIENIS A.; LAPRIE, 1986; GRAY J.; SIEWIOREK, 1991;

AVIZIENIS et al., 2004; BAUER; ADAMS, 2012; BAUER, 2011; MACIEL et al., 2012b; AC02762480, 1987). As melhorias no dispositivo aumentaram a disponibilidade do sistema do computador com o passar do tempo. Anteriormente, em 1980, sistemas de computador bem executados ofereciam 99% de disponibilidade. Isso parece bom, mas corresponde a cerca de 100 minutos de indisponibilidade por semana. Essas interrupções podem ser aceitáveis para sistemas classificados como não críticos. Por outro lado, os aplicativos de missão crítica e *online* não podem tolerar esse tempo de indisponibilidade. Eles exigem sistemas de alta disponibilidade que oferecem 99,999% de disponibilidade: no máximo cinco minutos de interrupção do serviço por ano. A disponibilidade é adimensional e expressada por porcentagem de 0% a 100%, ou utilizando uma escala de 0 a 1. Um sistema que possua a disponibilidade de 99,999%, ou 0,99999, pode ser classificado como um sistema de **cinco 9's**. A Tabela 2, adaptada do livro (BAUER, 2011), mostra a relação entre a quantidade de 9's e o tempo de indisponibilidade por ano.

Tabela 2 – Relação entre disponibilidade e tempo de indisponibilidade

Números de 9's	**Disponibilidade**	**Disponibilidade (%)**	**Tempo anual de indisponibilidade**
1	0,9	90 %	36.5 dias
2	0,99	99 %	3.65 dias
3	0,999	99.9 %	8.76 horas
4	0,9999	99.99 %	52 minutos
5	0,99999	99.999 %	5 minutos
6	0,999999	99.9999 %	30 segundos
7	0,9999999	99.99999 %	3 segundos

Baseada em: BAUER, 2011

A Equação 2.1 define a disponibilidade de estado estacionário de um sistema (WANG; TRIVEDI, 2005), onde $\boldsymbol{E_{up}}$ corresponde ao tempo de atividade do sistema e $\boldsymbol{E_{down}}$ é o sistema tempo de indisponibilidade.

$$A = \frac{E_{up}}{E_{up} + E_{down}}. \tag{2.1}$$

O tempo de atividade do sistema equivale ao Tempo Médio de Falha (MTTF) e o tempo de indisponibilidade, ao Tempo Médio de Reparo (MTTR) (WANG; TRIVEDI, 2005; SCHNEEWEISS,

2012; MACIEL et al., 2012b). Então, o cálculo da disponibilidade também pode ser expresso pela Equação:

$$A = \frac{MTTF}{MTTF + MTTR}. \tag{2.2}$$

Ao utilizarmos os valores de MTTF e MTTR para realizar o cálculo da disponibilidade, o resultado será sempre um número entre 0 e 1. O MTTF do sistema pode ser calculado pela Eq. 2.3, onde $\boldsymbol{R(t)}$ é a confiabilidade desse sistema em função do tempo decorrido.

$$MTTF = \int_0^\infty R(t)\ . \tag{2.3}$$

Sendo a taxa ($\boldsymbol{\gamma_i}$) inversamente proposicional ao valor do MTTF, então temos:

$$\gamma_i = \frac{1}{MTTF}\ . \tag{2.4}$$

Para os sistemas que operam em série e possuam as taxas $\boldsymbol{\gamma_i}$ constantes e iguais, o MTTF pode ser calculado com a Equação 2.5,

$$MTTF = \frac{1}{\sum_{i=1}^{n} \gamma_i}\ , \tag{2.5}$$

onde $\boldsymbol{n}$ é o número de componentes e $\boldsymbol{\gamma_i}$ a taxa (MACIEL, 2022b). Em situações em que todos os γ_i são iguais, pode-se chamar de $\boldsymbol{\gamma}$, então temos:

$$MTTF = \frac{1}{n \times \gamma}\ . \tag{2.6}$$

Já para os sistemas que funcionam em paralelo com taxas de falha iguais ($\boldsymbol{\gamma}$) e constantes, temos:

$$MTTF = \frac{1}{\gamma} \times \sum_{i=1}^{n} \times \frac{1}{i}\ . \tag{2.7}$$

Em sistemas *K-out-of-N* (KooN), com taxas de falha iguais ($\boldsymbol{\gamma}$) e constantes, temos:

$$MTTF = \frac{1}{\gamma} \times \sum_{i=k}^{n} \times \frac{1}{i} \tag{2.8}$$

Onde $\boldsymbol{k}$ se referencia ao número de componentes que precisam estar ativos para que o sistema esteja funcional e $\boldsymbol{n}$ são as quantidades de componentes. As explicações sobre os termos *série, paralelo e KooN*, estão apresentados na Subseção 2.5.1.

O valor estimado do MTTF ($\widehat{\boldsymbol{MTTF}}$) pode ser encontrado através da Equação 2.9 (MACIEL, 2022b), onde se utiliza da razão entre o somatório dos Tempo de Falha (TTF) e a quantidade de falhas $\boldsymbol{n}$. Neste estudo, os valores utilizados nos cálculos de medidas estimadas, foram coletados através do monitoramento do ambiente estudado.

$$\widehat{MTTF} = \frac{\sum_{i=1}^{n} ttf_i}{n} \, . \tag{2.9}$$

Utilizando o mesmo raciocínio da Equação 2.9, pode-se estimar o valor do MTTR ($\widehat{\boldsymbol{MTTR}}$) através da Equação 2.10.

$$\widehat{MTTR} = \frac{\sum_{i=1}^{n} ttr_i}{n} \, . \tag{2.10}$$

A Equação 2.11 fornece uma maneira de calcular o MTTR a partir dos valores de MTTF, disponibilidade e indisponibilidade (UA $= 1 - A$) (WANG; TRIVEDI, 2005; SCHNEEWEISS, 2012). Desta forma, pode-se encontrar o MTTR seguido a Equação,

$$MTTR = MTTF \times \left(\frac{UA}{A}\right). \tag{2.11}$$

Usando o mesmo raciocínio, pode-se encontrar o MTTF se utilizando uma variação da Equação 2.11. Assim, o MTTF também pode ser encontrado pela Equação

$$MTTF = \frac{A \times MTTR}{UA} \, . \tag{2.12}$$

A disponibilidade também pode ser representada pelo número de noves ($\#\,\boldsymbol{9's}$), que é estimado por

$$\#9s = -\log(1 - A), \tag{2.13}$$

e o Tempo de Indisponibilidade (DT) pode ser calculado pela Equação 2.14, onde $\boldsymbol{t}$ representa o período de observação (anual, mensal, diário, etc), podendo ser representado em horas, minutos, segundos, etc.

$$DT = UA \times t. \tag{2.14}$$

2.3.2 Redundância

Existem várias técnicas que podem ser utilizadas com o intuito de aprimorar a disponibilidade de ambientes em nuvem, como criação de clusters, utilização da virtualização, dentre outras (LIU; SONG, 2003; YEOW et al., 2010; WALTERS et al., 2008). A maioria destas técnicas são baseadas na utilização do mecanismo de redundância, ou seja, replicação de componentes garantindo a segurança e a disponibilidade dos serviços, mesmo em caso de defeito em algum componente.

Os mecanismos de redundância possuem componentes, também conhecidos como nós, que podem ser classificados como ativo/ativo e ativo/passivo. O mecanismo de redundância do tipo ativo/ativo se utiliza dos componentes - primário e secundário - trabalhando em conjunto para realizar o balanceamento de carga. Quando um dos componentes apresentar defeito, o outro assume a responsabilidade pelo serviço, recebendo todas as solicitações de acesso (BAUER; ADAMS; EUSTACE, 2011). Já o mecanismo que funciona de forma ativo/passivo se utiliza dos mesmos componentes - primário e secundário -, porém, apenas o primário é responsável por prover o serviço, enquanto o componente secundário fica em espera, pronto para atuar quando necessário. Caso o componente primário apresente defeito, o secundário assume a função de responsável pelo serviço (BAUER; ADAMS; EUSTACE, 2011).

O mecanismo de redundância ativo/passivo possui três técnicas bastante utilizadas em infraestrutura em nuvem. São elas: *Cold Standby, Hot Standby e Warm Standby* (MACIEL; DANTAS; JÚNIOR, 2019; BAUER; ADAMS; EUSTACE, 2011; MACIEL et al., 2012a; MELO et al., 2018). Na técnica *Cold Standby*, o componente secundário se encontra desligado em *standby*, sendo acionado apenas se o componente principal apresentar defeito. O ponto positivo dessa técnica é que o nó secundário tem baixo consumo de energia e não sofre desgaste dos componentes. Por outro lado, esta técnica necessita de um tempo significativo para retornar ao estado funcional, aumentando a indisponibilidade do sistema.

A técnica *Hot Standby* pode ser considerada a mais transparente das três apresentadas. Nesta configuração, apenas o componente principal recebe a carga de acesso, porém, o componente secundário está ligado e recebendo o sincronismo dos dados, tornando-o cópia idêntica ao componente principal. Caso o componente principal apresente defeito, automaticamente o secundário assume a função ativa respondendo pelo serviço. O ponto positivo desta técnica é que a mudança de equipamento afeta minimamente a disponibilidade do serviço. Por outro lado, existe um desgaste maior dos equipamentos por estarem todos sempre ligados e

recebendo algum tipo de carga de trabalho.

A técnica *Warm Standby* equilibra os custos e o tempo de recuperação das técnicas *Cold* e *Hot Standby*. O nó secundário está em espera, mas não completamente desligado. Portanto, pode ser ativado mais rápido do que na técnica *Cold Standby*. O nó secundário está parcialmente sincronizado com o nó operacional, de modo que os usuários podem perder algumas informações no momento exato da transição para o nó secundário.

2.4 BOOTSTRAPPING E ANÁLISE DE SENSIBILIDADE

Bootstrapping é um procedimento estatístico que realiza reamostragem de um único conjunto de dados para criar amostras simuladas. Esse método é comumente utilizada para construir intervalos de confiança, calcular erros padrão e realizar testes de hipóteses para vários tipos de estatísticas de amostra (HILLIS; BULL, 1993). Outra funcionalidade do *bootstrapping* é suprir uma determinada insuficiência de dados, já que possui a capacidade de gerar estatísticas populacionais por amostragem de um conjunto de dados (DIXON, 2006).

Já a AS, pretende quantificar as variações dos parâmetros nos resultados calculados. Termos como influência, importância, classificação por significância e dominância, estão relacionados à análise de sensibilidade. A AS pode ser considerada um método formal de avaliação de dados e modelos para determinar quais fatores são mais influentes em um sistema (HAMBY, 1994).

Uma abordagem típica para avaliação de modelo envolve a realização de cálculos com valores de parâmetros de entrada específicos para produzir valores de saída e gráficos de dispersão (CACUCI; IONESCU-BUJOR; NAVON, 2005). Assim, o objetivo científico da análise de sensibilidade não é confirmar noções preconcebidas, como sobre a importância relativa de *inputs* específicos, mas descobrir e quantificar as características mais importantes dos modelos sob investigação (MATOS et al., 2015; GHANEM; HIGDON; OWHADI, 2017).

A metodologia sistemática utilizada para realizar a AS, analisa as mudanças na distribuição de dados e seu impacto no sistema. Primeiramente, identifica qual componente do sistema tem a interferência mais significativa na métrica final (MATOS et al., 2017). Quando uma ligeira mudança em um componente do sistema resulta em uma variação significativa na métrica final, sabe-se que o sistema é suscetível a este parâmetro (KOPITTKE; FILHO, 2000).

Algumas técnicas de análise de sensibilidade foram desenvolvidas e relatadas na literatura (MELO et al., 2017; MATOS et al., 2015). Neste trabalho, é empregada uma técnica de diferença

percentual para calcular o índice de sensibilidade em relação com a função da disponibilidade. Esta função é composta por vários parâmetros $A = \{p_1, p_1, p_1, ..., p_n\}$, onde $\boldsymbol{p}$ representa um parâmetro necessário para a obtenção da disponibilidade. Por exemplo, para alcançar a disponibilidade do banco de dados, pode-se utilizar os parâmetros MTTF e MTTR dos componentes necessários para que o serviço esteja disponível. Assim, a função da disponibilidade possui os parâmetros $MTTF_{vm}$, $MTTR_{vm}$, $MTTF_{bd}$ e $MTTR_{bd}$.

A técnica utilizada para AS, implica em realizar alterações individuais, em todos os parâmetros necessários para a obtenção da disponibilidade. Cada alteração do parâmetro provoca um alteração na métrica da disponibilidade. A AS irá identificar individualmente qual parâmetro possui um maior impacto na métrica desejada. Ainda utilizando como exemplo o banco de dados, pode-se alterar o parâmetro $MTTF_{vm}$ 10 vezes (esta quantidade foi pré-definida pelo autor, podendo ser qualquer valor). Em cada uma das alterações, se obtém um valor de disponibilidade diferente. O mesmo se realiza para o parâmetro $MTTR_{vm}$ e consequentemente até finalizar todos os parâmetros.

Ao finalizar todas as alterações desejadas para o valor do parâmetro e obtenção dos valores da métrica desejada, em cada uma das alterações, pode-se encontrar o índice de sensibilidade do parâmetro, a partir da Equação 2.15.

$$S_{p(A)} = \frac{A_{p(max)} - A_{p(min)}}{A_{p(min)}} \ , \tag{2.15}$$

onde $\boldsymbol{S}$ é o índice de sensibilidade que está sendo calculado, $\boldsymbol{p}$ se refere ao parâmetro que sofreu alteração, $\boldsymbol{A}$ é a métrica avaliada, $\boldsymbol{A_{p(max)}}$ é o maior valor encontrado com a alteração do parâmetro $\boldsymbol{p}$ e $\boldsymbol{A_{p(min)}}$ se refere ao menor valor obtido com a variação do parâmetro $\boldsymbol{p}$. Quanto maior o valor do índice de sensibilidade, maior o impacto deste parâmetro na métrica desejada. Ao se realizar este procedimento para todos os parâmetros que compõem a função da disponibilidade, pode-se construir uma classificação da AS. Essa classificação melhora a previsibilidade de maior disponibilidade (LENHART et al., 2002).

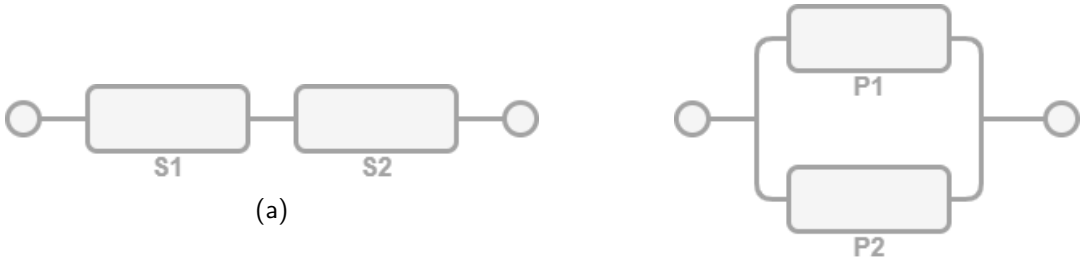
2.5 MODELAGEM

Esta seção apresenta duas estruturas de modelos de disponibilidade amplamente utilizadas para avaliar sistemas operando em nuvem. Primeiramente, são apresentados os conceitos básicos sobre o modelo RBD, em seguida, fundamentos de SPN. Neste trabalho, os modelos

em RBD são utilizados como o primeiro nível de uma modelagem hierárquica, utilizados para extrair os tempos de MTTF e MTTR de determinados componentes. Já os modelos em SPN representam o segundo nível hierárquico, sendo um modelo de alto nível com capacidade de realizar simplificações. Ambos são utilizados para construção de modelos de disponibilidade.

2.5.1 Modelo em RBD

O Diagrama RBD (KUO; ZUO, 2003; ČEPIN, 2011; WANG et al., 2004) é uma técnica de análise gráfica que representa sistemas ou componentes como blocos e, seus relacionamentos funcionais, como conexões entre eles (GUO; YANG, 2007). Os blocos no RBD são posicionados dependendo das suas funções no sistema que está sendo modelado. Existem conexões seriais e paralelas. Na conexão serial, representada na Figura 7a, existe uma representação lógica *AND*. Isto significa que todos os componentes precisam estar operacionais para que o sistema esteja disponível (KIM, 2011). Caso um bloco apresente defeito, todo o sistema estará em falha. A representação do cálculo da disponibilidade no RBD em série está grafada na Equação 2.16.



(a) (b)

Figura 7 – Estruturas de modelos em RBD.

$$A_{(s)} = \prod_{i=1}^{n} A_{i(s)}. \tag{2.16}$$

Onde o termo $\boldsymbol{A_{(s)}}$ equivale à disponibilidade do sistema em questão, a termo $\boldsymbol{n}$ equivale às quantidades de componentes do sistema e $\boldsymbol{A_{i(s)}}$ é o valor da disponibilidade individual de cada componente.

A segunda possibilidade de interação entre os blocos é a conexão paralela, exposta na Figura 7b. Esta conexão possui uma representação lógica *OU*, significando que pode existir redundância de equipamentos (KIM, 2011). Um componente pode apresentar defeito e este fato não caracteriza necessariamente uma interrupção do sistema. A representação do cálculo em paralelo se refere à Equação 2.17, onde o termo $\boldsymbol{A_{(p)}}$ equivale à disponibilidade do sistema em

questão, $\boldsymbol{n}$) são as quantidades de componentes do sistema e $\boldsymbol{A_{i(p)}}$ é o valor da disponibilidade individual de cada componente.

$$A_{(p)} = 1 - \prod_{i=1}^{n}(1 - A_{i(s)})\,. \tag{2.17}$$

A última configuração possível na modelagem em RBD é KooN, que representa o número de componentes que devem estar no estado operacional para que o bloco esteja operacional (MACIEL, 2022b). Pode ser encontrado através da Equação 2.18.

$$A_{KooN} = \sum_{i=k}^{n} \binom{n}{i} Ac^i \times (1 - Ac)^{n-i}\,. \tag{2.18}$$

Onde o $\boldsymbol{K}$ representa a quantidade mínima de componentes que precisam estar operacionais, o $\boldsymbol{N}$ simboliza a quantidade total de equipamentos e a expressão $\boldsymbol{Ac}$ representa a disponibilidade individual do componente. Todas essas condições afetam diretamente a disponibilidade do sistema e podem ser calculadas de acordo com princípios probabilísticos. O cálculo da disponibilidade de um sistema ou subsistema depende da redundância de cada componente.

2.5.2 Modelo em SPN

A *Petri Net* (PN) é um conceito introduzido em (PETRI, 1966). Este conceito é um paradigma visual para a descrição formal das interações lógicas entre as partes ou o fluxo de atividades em sistemas complexos (MURATA, 1989; MACIEL; BARROS; ROSENSTIEL, 1999). Alguns trabalhos podem ser encontrados em (PETERSON, 1977; PETERSON, 1981). Inicialmente, a PN não utiliza temporização em sua modelagem. No entanto, precisamos desse recurso para realizar análises de confiabilidade e disponibilidade. Portanto, este trabalho utiliza uma extensão conhecida como SPN que foi estudada extensivamente (MOLLOY, 1982; MARSAN; CHIOLA, 1986; GERMAN, 2000; MACIEL; BARROS; ROSENSTIEL, 1999), onde tempos podem ser associados às transações.

De acordo com (MARSAN; CONTE; BALBO, 1984; MOLLOY, 1981), os SPNs são obtidos associando a cada transição em uma PN um tempo de disparo exponencialmente distribuído. Os autores mostraram que os SPNs correspondem a *Continuous-Time Markov Chain* (CTMC) devido à propriedade sem memória da distribuição exponencial dos tempos de disparo. Portanto, as marcações SPN correspondem aos estados CTMC.

As SPNs são amplamente usadas em modelos probabilísticos para análise de desempenho. Esta modelagem é uma ferramenta útil para analisar sistemas de computador, pois permite que as operações do sistema sejam descritas com precisão através de um gráfico, que no que lhe concerne, se traduz em um modelo Markoviano útil para obter estimativas de desempenho (MACIEL; DANTAS; JÚNIOR, 2019). O modelo SPN permite o cálculo das probabilidades de estado estacionário e a análise de medidas de desempenho como atraso médio e *throughput* médio. Toda essa análise é realizada usando o modelo de Markov equivalente (MARSAN et al., 1998).

O modelo SPN funciona com lugares e transições. Os lugares representam estados ou condições do sistema, enquanto as transições representam eventos, que podem ou não causar mudanças no estado do sistema. Outros componentes são arcos e *tokens*. Sempre que existir *token* em um determinado lugar, sinaliza que aquele estado se encontra ativo.

Os componentes utilizados neste estudo estão representandos na Figura 8. Uma breve descrição dos componentes segue:

Figura 8 – Objetos básicos do modelo SPN

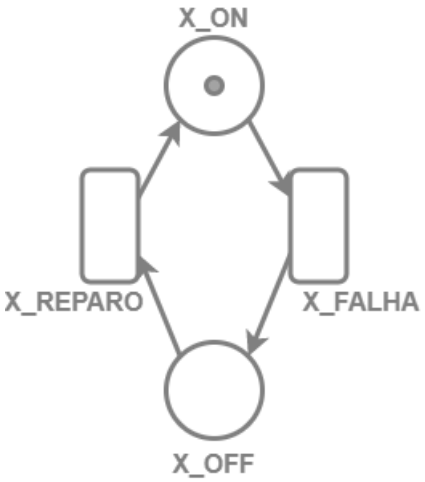
- **Lugar ativo** representa um estado ou condição do sistema. O lugar está ativo e habilitado quando existem *tokens*;
- **Lugar inativo** também representa um estado ou condição do sistema, porém, não está habilitado por não possuir nenhum *token*;
- **Transição temporizada** representa uma ação que possui um determinado tempo para ser executada;
- **Transição imediata** representa uma ação que não possui tempo para ser executada, porém, se torna habilitada com condições específicas. Quando este tipo de transição está habilitada, precisa ser executada imediatamente;
- **Arco de transição** é a conexão entre componentes do modelo. Os arcos conectam lugares e transições, e são responsáveis por consumir e gerar *tokens*, dependendo de

onde estiver conectado;

- **Arco inibidor** é uma conexão que não permite a habilitação de uma transição;
- ***Token*:** Representa a quantidade de itens em um lugar;
- **Expressão de guarda** é a regra que precisa ser atendida para habilitar uma determinada transição.

A modelagem básica de uma SPN está exposta na Figura 9. Esta estrutura de modelo é utilizada para representar servidores, aplicação, banco de dados e até realizar uma sintetização de estruturas mais complexas. Em outras palavras, tudo que pode ser representado por dois estados, operacional ou falha. Este modelo permite o cálculo da disponibilidade dos componentes desejados.

Figura 9 – Modelo básico em SPN



No modelo SPN apresentado na Figura 9, a representação do rótulo X identifica o subsistema que está sendo modelado. Exemplificando, caso o modelo represente um banco de dados, a marcação X será substituída por DB. Desta forma, existem lugares identificados como BD_ON e DB_OFF, para representar, respectivamente, o estado operacional e de falha do subsistema Banco de dados. O que definirá o estado do modelo é a localização do *token*.

As transições temporizadas identificadas no modelo como X_FALHA e X_REPARO representam as ações que levam a troca de estado do sistema. Elas possuem o valor do MTTF e MTTR, respectivamente. Esta representação permite realizar cálculos para identificar a disponibilidade do sistema, conforme apresentado na Equação 2.19. O termo $A_{(s)}$ representa a disponibilidade do sistema. O termo P representa a probabilidade de acontecer algo. O termo

(#) representa a quantidade de tokens em um determinado lugar. A representação X_ON informa qual lugar está sendo referenciado e o termo (>0) representa a condição da Equação. Com isso, pode-se calcular a disponibilidade do nosso modelo, identificando a probabilidade de existir mais de um token no estado/lugar X_ON.

$$A_{(s)} = P\{\#X\_ON > 0\} \tag{2.19}$$

O modelo SPN possui um fluxo de transição de tokens que sinaliza a falha ou reparo do sistema modelado dependendo do lugar onde possui *token*. O comportamento do modelo é apresentado na Figura 10.

Figura 10 – Fluxo do modelo genérico em SPN

Sistema no estado operacional está representado na Figura 10a, inicialmente o sistema está funcional, pois possui um token no lugar X_ON. A única transição ativa é X_FALHA pois está conectada ao único lugar possível de retirada de tokens. Quando a transição é acionada, o token é consumido pelo arco de transição e gerado no lugar X_OFF, identificando o sistema no estado de falha. A transição temporizada possui um valor referente ao MTTF do sistema ou componente representado.

Sistema no estado em falha é representado na Figura 10b. Este modelo indica que o sistema está inoperante, por existir *token* no lugar *X_OFF*. A única transição ativa é X_REPARO por estar conectada ao único lugar possível de retirada de *tokens*. Quando a transição é acionada, o token é consumido do lugar X_OFF e gerado no lugar X_ON. Neste momento, o modelo retorna ao estado operacional. O valor desta transição se refere ao MTTR do componente ou sistema modelado.

2.6 CONSIDERAÇÕES FINAIS

Este capítulo apresenta a fundamentação teórica básica para o melhor entendimento do leitor sobre os principais tópicos que compõem esta dissertação. Os conceitos básicos sobre estrutura de centro de dados, computação em nuvem, dependabilidade e modelagem, permitem o entendimento do ambiente e dos estudos de casos que apoiam a proposta de melhoria do ambiente estudado.

3 TRABALHOS RELACIONADOS

Este capítulo apresenta um conjunto de pesquisas acadêmicas relacionadas com o tema desse trabalho. Essas pesquisas foram selecionadas através de um levantamento bibliográfico com o intuito de coletar e analisar estudos que abordassem técnicas e soluções para melhorar a disponibilidade de sistemas hospedados em nuvem que utilizem CD distribuídos. Foram realizadas pesquisas com o intuito de identificar temas abordados nesse estudo, utilizando um filtro de tempo de seis anos, para serem apresentados estudos mais focados em inovações tecnológicas e de pesquisas mais recentes. Alguns dos trabalhos aqui apresentados possuem um tempo maior de publicação, mas foram incorporados pela sua constante citação em trabalhos acadêmicos ou pela importância de conhecimento retirado deles. Os temas pesquisados foram:

- Infraestrutura do Centro de dados;
- Centro de dados redundante;
- Alta disponibilidade/*Cluster*/Virtualização;
- Modelos estocásticos.

Após pesquisas refinadas sobre os temas listados acima, foram coletados 140 trabalhos. Em seguida, na fase de triagem, buscou-se excluir os estudos que estavam fora do escopo da pesquisa. Após esse filtro, foram selecionados 50 trabalhos que estavam relacionados aos temas macros e minimamente alinhados com o tema deste trabalho, em seguida a análise criteriosa, identificaram-se os trabalhos voltados diretamente à disponibilidade de sistemas, que se utilizam ou não, da redundância de centro de dados como solução de aprimorar a disponibilidade dos sistemas. Desta forma, foram selecionados 18 trabalhos acadêmicos referenciados abaixo.

Callou et al. (CALLOU; ANDRADE; FERREIRA, 2019), apresentam análise de custo, sustentabilidade e disponibilidade de um CD. Para atingir o objetivo do trabalho, foram utilizados modelos estocásticos em RBD, SPN e Modelo de Fluxo de Energia (EFM). Os autores desenvolveram estudo visando economia energética, vislumbrando minimizar os custos elevados para manter a alta disponibilidade. O trabalho expõe modelos em RBD e SPN para definir a disponibilidade do CD e propor configurações de ambiente que visam melhorias na estrutura. Segundo o estudo, foi possível um acréscimo de 70% de disponibilidade com o aumento de 15% dos custos de equipamentos. O trabalho apresentou modelos de disponibilidade bem

estruturados que serviu como base de estudo para este trabalho. Porém, não tem o foco de apresentar redundância dos componentes de TI. Esta abordagem é natural por ter um foco em fornecimento energético.

Rosendo et al. (ROSENDO et al., 2018) apresentam um estudo onde indicam que 25% das falhas que provocam indisponibilidade nos serviços de um CD são causadas por interrupção energética. Visando ajudar na melhoria da disponibilidade dos CDs, o estudo utiliza a modelagem estocástica em RBD e SPN como técnica de estimativa de disponibilidade focando no fornecimento de energia e em serviços de TI. O estudo aborda a modelagem voltada para o fornecimento energético e subsistema em TI, utilizando CDs que possuem as classificações pela TIA (*Tier* I à IV). Os autores afirmam que usando componente redundante em sistemas de energia e TI, pode-se diminuir consideravelmente o tempo de indisponibilidade de um serviço em execução nos CDs. Alterando a arquitetura de um CD *Tier* I para o nível *Tier* IV, existe o ganho de 36,28%, que equivale a uma redução no tempo de indisponibilidade de 19,65 horas/ano. O trabalho apresenta conclusões bastante significativas na melhora da disponibilidade de serviços de TI, com embasamentos e modelos bem estruturados. Em um dos Estudos de Casos apresentados nesta dissertação, identificou-se a grande interferência da estrutura de CD na disponibilidade dos sistemas hospedados, porém, para resolução deste problema, está sendo apresentada uma abordagem diferente. É sabido da eficiência de aprimorar o CD para melhorar a disponibilidade, porém, será abordado a replicação de CD como estratégia de aprimoramento da disponibilidade do sistema hospedado. A estratégia diferente ocorre pela possibilidade de recuperação de desastre, replicação de dados, balanceamento de carga, dentre outras. O caminho distinto não exime a importância deste trabalho como base de conhecimento para melhoria da disponibilidade de um CD.

De acordo com Santos et al. em (SANTOS et al., 2017), um CD pode ser dividido em três subsistemas principais: sistemas de refrigeração, energia e infraestrutura de TI. Os autores informam que esses sistemas são independentes, mas que interrupções podem afetar as disponibilidades uns dos outros. O estudo propõe modelos em RBD e SPN para avaliar a disponibilidade de um serviço hospedado em um CD que utiliza infraestrutura de nuvem. O trabalho tem a abordagem de comparar componentes e disponibilidade do subsistema de TI, hospedados em CD classificados como *Tier* I e IV. As modelagens apresentadas são bem estruturadas, informando características de cada transição e suas respectivas guardas. Os autores concluem que entre os componentes mapeados e classificados como primordiais na funcionalidade do CD *tier* I, o MTTR do roteador de borda é o parâmetro que apresenta um maior impacto na

disponibilidade dos serviços. Ainda segundo os autores, um aumento de 2h no MTTR do roteador de borda equivale à redução da disponibilidade de 99,76% para 99,56%, com o resultado de indisponibilidade sendo aumentado de 21,02 para 38,55 horas anuais. Já na análise em um CD *tier* IV, como toda camada de conectividade de rede e armazenamento são redundantes, os parâmetros que apresentam maior impacto na disponibilidade são o MTTR e MTTF do servidor, respectivamente. O terceiro parâmetro que possui o maior impacto é o MTTR do roteador de borda. No estudo desse cenário, é constatado que um aumento de 2h no valor do MTTR do servidor equivale à redução da disponibilidade de 99,904% para 99,87%. O estudo também apresenta que existe um aumento considerável de disponibilidade do ambiente em um CD classificado como *Tier* I para um classificado como *Tier* IV. A melhora na disponibilidade varia de 99,781% para 99,903%, respectivamente, o que equivale à diminuição do tempo de indisponibilidade anual de 20,86 para 8,49 horas.

No trabalho (ROSENDO et al., 2019), Rosendo et al. realizaram uma abordagem de comparação de disponibilidade entre um CD convencional, chamado no trabalho de *Performance Optimization Data Center* (POD), e um CD virtual, chamado *Virtual Performance Optimization Data Center* (vPOD). Eles colheram informações dos ambientes utilizando *Application Programming Interface* (API) da *Redfish* para definir os tempos de MTTR e MTTF do ambiente e elaboraram modelos em RBD para cada cenário. Os autores utilizam a ferramenta Mercury para a construção dos modelos e realizam comparações utilizando entre dois a cinco servidores. A conclusão do trabalho expõe as vantagens no valor da disponibilidade e financeira da utilização do vPOD. O trabalho elabora uma pesquisa bem estruturada, mas não aborda modelagens como cadeia de *Markov* ou SPN que poderiam representar melhor a disponibilidade do ambiente, nem especifica componentes sensíveis que melhorariam os índices de disponibilidade, todavia, expõem que o conduzirão em trabalhos futuros.

O trabalho de Mendonça et al. (MENDONÇA et al., 2018), adota uma abordagem de injeção de falhas em um modelo em SPN, visando a avaliação da disponibilidade de um ambiente de CD distribuído que possui uma solução de Recuperação de Desastre (DR). O estudo é bem estruturado, apresentando modelos que contemplam cada categoria de DR, ativo/ativo e ativo/passivo, bem como diferentes formas de falhas dos componentes. Os autores afirmam que a utilização de uma solução de DR afeta positivamente a disponibilidade dos sistemas e reduz os custos operacionais. A presente dissertação também se utiliza da replicação de CD para aprimoramento da disponibilidade de sistemas, tendo sua principal diferença na identificação de qual o CD está ativo, assim como na análise e proposta de melhorias da infraestrutura

interna ao CD.

Os autores Nguyen, Kim, Park e Sou em (NGUYEN; KIM; PARK, 2016), apresentam um trabalho direcionado para um estudo de um CD tolerante a desastre. No estudo, foi criado um modelo utilizando *Stochastic Reward Net* (SRN), que visa modelar ambientes de CDs geograficamente distribuídos com o intuito de aprimorar a disponibilidade do CD. Os modelos apresentados são bastante completos e contemplam várias categorias de falhas, desastres e redundâncias internas e externas ao CD, utilizando dados reais para suas métricas. Algumas das características contempladas no modelo são: configuração de Alta Disponibilidade (HA) (ativo/ativo entre sites e ativo/passivo dentro de um site); técnicas de tolerância a falhas e desastres (migração de VM ao vivo entre hosts e entre sites); níveis de relação e dependências entre subsistemas (entre hosts e VMs, entre armazenamento de área de rede (NAS) e VMs, e entre os eventos de ocorrência de desastre e hosts e NAS); e possíveis falhas inesperadas durante a transmissão de dados entre data centers.

Como resultado, os autores apresentam que um Data Center Tolerante a Desastres (DTDC) possui um melhor índice de disponibilidade comparado a CD redundantes que não possuem sistema de tolerância a falhas. Foi identificado também que um centro de dados com Tolerância a Falha (FT) e outro com tolerância a desastre situado em um local com poucos eventos desastrosos (área mais segura) tem maior disponibilidade, mesmo com conectividade de rede de menor qualidade. Foi apresentado também que as falhas na transmissão de VM de longa distância (entre o servidor secundário e o CD) afetam significativamente, enquanto as falhas entre os CDs têm menos efeito sobre a disponibilidade do sistema. Por último, se identificou que o tamanho da VM interfere diretamente na disponibilidade do sistema, o impacto será agravado ou amenizado conforme o tamanho do link de dados entre os CDs.

Andrade et al. (ANDRADE et al., 2017) apresentam um estudo de disponibilidade para avaliar uma solução de *Disaster Recevery as a Service* (DRaaS) abordando assuntos como tempo de indisponibilidade e custos. Foi realizado também um estudo de sensibilidade para identificar quais parâmetros possuem maior impacto na disponibilidade. O foco do estudo é apresentar dados para que os gestores de desastres possam avaliar a estrutura e tomar decisões sobre a recuperação do serviço. O trabalho se utiliza de um cenário real para apresentar vários modelos bem estruturados em SPN. Estes modelos separam a infraestrutura do serviço da infraestrutura do CD. O resultado da modelagem e do estudo, apontam que uma solução de DRaaS possui uma melhora na disponibilidade e posteriormente, reduz o custo relacionados ao tempo de indisponibilidade. Sobre a análise de sensibilidade, os autores chegaram a uma

conclusão que as taxas de reparo e falha dos componentes da infraestrutura primária afetam mais intensamente o índice de disponibilidade do sistema. Foi constatado também que os ajustes finos nos componentes de recuperação de desastre possuem pouco ou nenhum efeito nas métricas após a solução de DRaaS está implantada.

No trabalho (MSEDDI et al., 2018), Mseddi et al. apresentam uma solução chamada CRANE. A solução apresenta um esquema de migração de réplica eficiente para sistemas de armazenamento em nuvem distribuídas, contemplando algoritmos de réplicas que facilitam o gerenciamento dos dados em ambientes geograficamente distribuídos. No trabalho em questão, foram realizados experimentos reais que apresentaram a desempenho do CRANE em conjunto com o OpenStack[1] em diferentes cenários. Estas soluções foram comparadas com outras soluções de mercado. O trabalho apresenta modelos matemáticos representando a solução e como contornar alguns cenários desfavoráveis, porém, não apresenta modelos estocásticos para expor os índices de melhora. Os resultados mostram, segundo os autores, que o CRANE possui desempenho abaixo do ideal em relação ao tempo de migração e um desempenho bom em relação à quantidade de dados transferidos. Informa também que conseguiu reduzir em até 60% o tempo de criação das réplicas e em 50% do tráfego da rede entre CDs, permitindo uma melhor disponibilidade de dados durante o processo de migração de réplicas.

O trabalho de Silva et al. (SILVA et al., 2018) permite aos projetistas de sistemas em nuvem identificar parâmetros impactantes da disponibilidade do serviço através de uma análise de sensibilidade paramétrica, utilizando CTMC. Os autores consideram a migração de máquinas virtuais, falhas de *hardware* e *software*, e desastre do ambiente. No trabalho, foi mostrado que a distância geográfica entre os CDs interfere diretamente na disponibilidade dos sistemas, principalmente pela latência de rede na migração das VMs. No estudo de sensibilidade, foi constatado que a taxa de falha de *hardware* e *software*, ocorrência de desastre e o tempo de migração das VMs são os principais parâmetros que interferem na disponibilidade dos sistemas. O estudo conduz uma análise detalhada e bem elaborada do sistema e um modelo bastante interessante de disponibilidade em CTMC, realizando cálculos, simulações e validações para comprovar a efetividade do modelo. Em contrapartida, não possuem dados do ambiente real nem realizou simulações, assuntos previstos nos trabalhos futuros.

Torquato et al. em (TORQUATO et al., 2019) apresentam um modelo hierárquico escalável em RBD e SRN para avaliação de disponibilidade em ambiente de Centro de Dados Virtual (VDC). Foram propostas oito arquiteturas distintas chegando até a utilização de mil e

[1] www.openstack.org

quatrocentas VMs no ambiente. Foi adicionado ao índice de disponibilidade e o cálculo de Capacidade Orientada a Disponibilidade (COA). Como resultado do estudo no cenário utilizado, se percebeu que a melhora da disponibilidade causada pela adição de elementos de arquitetura tem um limite. Quando utilizados ambientes com mais de mil VMs, a disponibilidade máxima não sofre mais interferência ao adicionarmos estes elementos. Chegando à conclusão de que nesse ponto, a adição de recurso físico ocasiona desperdício deste, por outro lado, aumenta significativamente o COA. O estudo propõe modelos bem elaborados e comprovados pelos cálculos apresentados, mas não considera a migração de VMs e envelhecimento do *software* e *hardware*, itens que afetam na disponibilidade do ambiente, porém, identificam como demanda futura.

Dhanujati et al. em (DHANUJATI; GIRSANG, 2018) apresentam um estudo sobre recuperação de desastre em CD, se utilizando de uma empresa real de eletricidade na Indonésia, com cerca de 37 milhões de usuários. O estudo realiza uma fundamentação e uma elaboração de infraestrutura necessária para que o serviço possa sempre estar operante, mesmo em caso de interrupção do CD principal, não perdendo nem as transações atuais. Mesmo o trabalho trazendo fundamentos elaborados para os cenários reais e propostos, não introduz a modelagem nem cálculos da disponibilidade. Também não apresenta a estratégia a ser utilizada no momento de desastre nem para a volta ao ambiente principal.

Alshammari et al. (ALSHAMMARI et al., 2017) apresentam discussões sobre o tema de recuperação de desastre e hospedagem dos serviços ofertados em nuvens, tanto em uma única nuvem como em vária. O trabalho examina estudos anteriores sobre os temas e provoca reflexões de pontos importantes trazidos por eles. O estudo percebe que existem pontos de relevante importância a serem solucionados e entendidos no ambiente em nuvem, como quantidade de armazenamento, tráfego dos dados entre as nuvens, período de detecção de falhas, segurança dos dados e *backups*. O estudo não foca em comprovar nem analisar modelos de disponibilidade, nem estrutura física do centro de dados.

Mesbahi et al. (MESBAHI; RAHMANI; HOSSEINZADEH, 2018) apresentam soluções em nuvem utilizando alta disponibilidade e propondo um *roadmap* de todos os estudos necessários para alcançar confiabilidade e disponibilidade classificada como aceitável. Apresenta a importância dos modelos combinatórios, de espaço de estados e hierárquicos. Já Torquato et al. (TORQUATO; UMESH; MACIEL, 2018) expõem o problema do envelhecimento do *software* e o impacto na disponibilidade do ambiente. Apresentam modelos de disponibilidade de SPN e usam técnicas de redundância como *Warm-Standby* e *Cold-Standby* para defender que a

migração a quente das máquinas virtuais contribui para o rejuvenescimento do *software*.

Melo et al. (MELO et al., 2017), avaliaram a COA de uma nuvem privada. O estudo foca em aproveitar melhor os recursos físicos da infraestrutura, e realiza a modelagem em RBD e SPN para subsidiar seus estudos. Os resultados foram satisfatórios, no entanto, eles não consideraram o uso de *clusters* virtuais e físicos ou a migração de máquinas virtuais para outros recursos físicos.

Matos et al. (MATOS et al., 2017) propõem um modelo de avaliação de disponibilidade hierárquica, representado por RBD, CTMC e SPN que correspondem a uma nuvem privada da arquitetura de ambientes baseados em Eucalyptus. Os modelos apresentados contemplam soluções de tolerância a falhas, como hosts redundantes em *hot standby* para alguns de seus componentes principais. Os autores elaboram matematicamente duas formas de execução de análise de sensibilidade. Segundo os autores, a análise de sensibilidade diferencial também pode ser usada para a disponibilidade e avaliação de desempenho de diferentes categorias de sistemas. A técnica é eficaz na análise de sistemas com muitos componentes e eventos enquanto outros métodos de análise de sensibilidade fornecem apenas uma visão parcial da influência de cada parâmetro.

Li et al. (LI et al., 2017) apresentam um método para modelar a confiabilidade do serviço referente à infraestrutura de TI com base na teoria das filas e na teoria dos grafos, e também propõe seu cálculo quantitativo e método de avaliação baseado no método de Monte Carlo. Os autores identificam que um CD em nuvem consiste em três infraestruturas principais: a infraestrutura de energia, a infraestrutura de resfriamento e a infraestrutura de TI. Foi utilizada uma infraestrutura de CD ativo/ativo, tendo o CD 1 e CD 2 ofertando serviços juntos e sendo um *backup* do outro. Os autores segmentam a modelagem de confiabilidade em duas partes: a modelagem da etapa de solicitação e a modelagem da etapa de execução. A etapa de solicitação é processada pelo Gerenciador de Sistemas de Nuvem (CMS) e a etapa de execução é distribuída entre os CDs, sendo atendidas pelos equipamentos de TI. Na modelagem, foram considerados várias categorias de falhas que possuem a capacidade de influenciar a confiabilidade do serviço em todo o processo de atendimento. Foram considerados, também, a distribuição do tempo de vida de cada nó, tendo um novo modelo de grafo do estágio de execução baseado na topologia física do CD. Os autores também executam uma análise de sensibilidade considerando a variação de fatores internos e externos, assim como a identificação de alguns fatores que possuem uma alta sensibilidade.

As conclusões do trabalho indicam que efetuar a abordagem de modelagem com foco em

teorias de filas e grafos, juntamente com a análise de sensibilidade se fez eficaz na identificação de parâmetros sensíveis à confiabilidade. Foi identificado também que a quantidade de componentes (equipamentos no CD), os *schedulers* (funcionalidade do CMS) e o tamanho máximo do *buffer* são parâmetros mais sensíveis à confiabilidade, divergindo da rede entre os CDs, que obteve o menor índice de sensibilidade à confiabilidade. O trabalho não considera a degeneração dos componentes físicos e não considera a virtualização de *software*, amplamente utilizada em ambientes em nuvem, porém serão temas abordados em trabalhos futuros.

Pan et al., no trabalho (PAN; HU, 2014), identificam os principais fatores que influenciam na confiabilidade da computação em nuvem e seus modos de falha. Nesta análise, foi percebido que os principais modos de falhas variam conforme o sistema utilizado na computação em nuvem (IaaS, PaaS ou SaaS), porém, compartilham de um ponto em comum, o CMS. O trabalho em questão também identifica que para melhorar a confiabilidade do sistema em computação, a redundância, a replicação e o *checkpoint* são mecanismos mais utilizados. Todo o estudo foi realizado com a ferramenta CloudSim[2], desenvolvido pelo Laboratório Clouds da Universidade de Melbourne. Esta ferramenta foi utilizada para desenvolver simulações de um sistema de computação em nuvem virtualizado, possibilitando que os pesquisadores efetuassem experimentos em infraestruturas de computação em nuvem. A conclusão do trabalho induz que a melhor forma de se analisar a confiabilidade de sistemas em nuvem é a simulação, justificando ser uma solução razoável e econômica com muitas vantagens em comparação com métodos analíticos que se utilizam de modelagem matemática. A simulação realizada se utiliza de métodos de infeção de falhas para identificar eventos de falhas e recuperação do CD. A abordagem deste trabalho difere da nossa, porém, traz pontos importantes para análise de qual método pode ser mais eficiente para análise de confiabilidade de sistemas complexos como os que se utilizam de computação em nuvem.

3.1 VISÃO GERAL

A Tabela 3 apresenta uma comparação entre os trabalhos relacionados e a presente dissertação em termos de principais contribuições. Os pontos analisados foram: a existência de informações e análises sobre a infraestrutura do CD; a existência de modelagens que representam a infraestrutura do ambiente e o sistema propriamente dito; realização de uma análise de sensibilidade que identifique os pontos mais sensíveis do ambiente; a existência de modelos

[2] http://www.cloudbus.org/cloudsim/

que representem o CD único e na configuração de redundância; por último, a existência de mecanismos que possam identificar qual o CD está ativo e tendo seus recursos físicos sendo utilizados.

Tabela 3 – Comparação entre os trabalhos relacionados mais relevantes.

Autores	***Análise da infraestrutura do CD***	***Modelagem da infraestrutura***	***Modelagem do sistema***	***Análise de sensibilidade***	***Modelo de CD único***	***Modelo de CD Redundante***	***Modelo de controle do CD Ativo***
(CALLOU; ANDRADE; FERREIRA, 2019)		✓	✓				
(ROSENDO et al., 2018)	✓	✓		✓	✓		
(SANTOS et al., 2017)	✓	✓	✓	✓	✓		
(MENDONÇA et al., 2018)			✓		✓	✓	
(NGUYEN; KIM; PARK, 2016)		✓	✓		✓	✓	✓
(ANDRADE et al., 2017)			✓		✓	✓	
(SILVA et al., 2018)				✓	✓	✓	
(TORQUATO; UMESH; MACIEL, 2018)		✓	✓		✓		
(MELO et al., 2017)		✓	✓	✓			
(MATOS et al., 2017)		✓	✓	✓	✓		
(LI et al., 2017)		✓	✓		✓	✓	
Esta dissertação	✔	✔	✔	✔	✔	✔	✔

Como pode-se perceber na comparação, esta dissertação se diferencia dos demais trabalhos previamente apresentados em alguns aspectos, pois realiza análise de disponibilidade de um sistema hospedado em uma nuvem privada, considerando aspectos internos e externos ao CD. O trabalho também apresenta avaliação do CD onde o sistema estudado está hospedado. Além disso, indica os pontos que possuem maior impacto na disponibilidade através da análise de sensibilidade da estrutura do sistema. Com base nos modelos de disponibilidade propostos, foi desenvolvida análise da utilização de sites redundantes para o ambiente estudado, apresentando

uma estrutura de controle que visa identificar qual CD se apresenta ativo. Com estes pontos elencados, é possível obter-se auxílio na tomada de decisão estratégica referente a configuração adequada na utilização de sites redundantes.

Também pode-se concluir que a maioria dos trabalhos relacionados foca na disponibilidade do sistema, considerando a estrutura interna do CD ou apresentam análises referentes à redundância de CDs, em distinção deste trabalho que traz para análise os dois cenários. Outro aspecto importante é que os trabalhos que apresentam estudos sobre redundância de CD, não se utilizam de estruturas que possam gerenciar qual o CD está ativo. Este ponto é de extrema importância para identificar qual CD está tendo seus recursos físicos utilizados. Outro aspecto de divergência é que muitos trabalhos recorrem a modelos de simulação para contemplar o estudo de disponibilidade. Neste trabalho se utiliza análise estacionária para buscar o valor da disponibilidade do ambiente. Como último ponto de divergência, foi elaborado para este trabalho metodologia que possibilite melhor entendimento, reprodução e continuidade deste estudo.

3.2 CONSIDERAÇÕES FINAIS

Este capítulo apresenta os principais trabalhos relacionados a esta dissertação. Se faz necessário ressaltar que a presente lista não é exaustiva e que muitos outros trabalhos relacionados podem ser encontrados em mecanismos de busca. No entanto, ressalta-se que o foco da maioria dos estudos é a análise de disponibilidade utilizando CD redundantes ou da estrutura da aplicação, não focando na conexão entre estes pontos e o controle de qual CD está ativo. Estes pontos, juntamente com a falta de apresentação da metodologia utilizada para o correto entendimento e reprodução do estudo, são os principais aspectos divergentes com as contribuições da presente dissertação.

4 METODOLOGIA

Este capítulo apresenta a metodologia utilizada no estudo, sendo descritos os passos realizados na construção da proposta de melhoria da disponibilidade do sistema estudado. O principal objetivo deste capítulo é possibilitar replicações futuras dos passos adotados, facilitando aprimoramentos da disponibilidade do sistema estudado ou de outros sistemas que possuam estruturas físicas e lógicas semelhantes (LEE, 1989).

A metodologia apresentada a seguir descreve os processos utilizados na construção do ambiente proposto. Estes processos avaliam e constroem modelos da infraestrutura utilizada pelo sistema, assim como propõem estruturas alternativas que visam aprimorar a disponibilidade do sistema.

4.1 METODOLOGIA PARA AVALIAÇÃO DE DISPONIBILIDADE

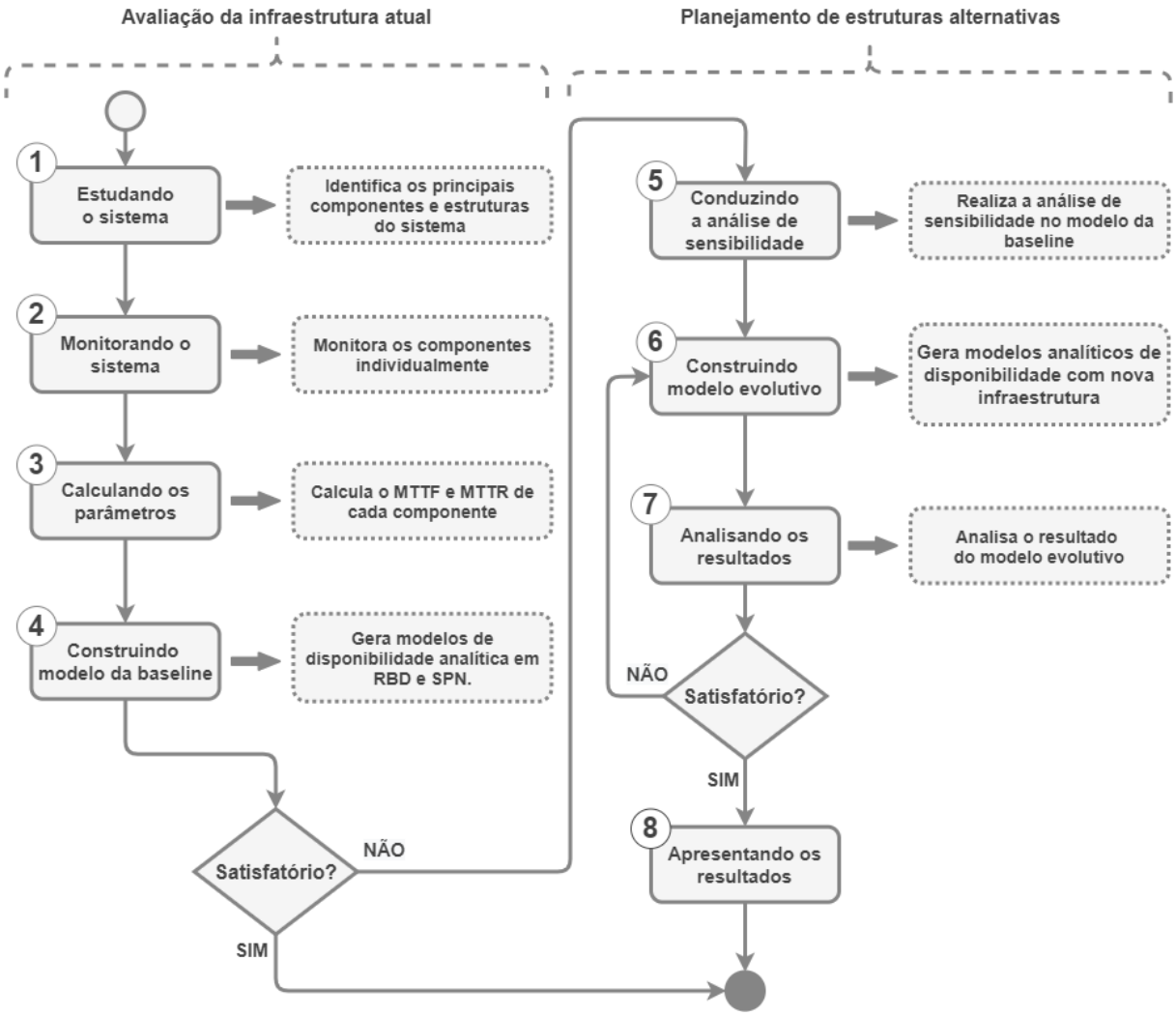
Esta seção apresenta uma visão da metodologia deste trabalho. Efetivamente a metodologia é dividida em duas etapas. A primeira etapa descreve todos os processos utilizados para a avaliação do sistema atual, apresentada na Subseção 4.1.1. A segunda etapa descreve os processos que possibilitam a avaliação de estruturas alterativas que visam a melhoria da disponibilidade do sistema atual, apresentada na Subseção 4.1.2. A visão geral da metodologia está apresentada na Figura 11.

Os processos são retratados por macro-atividade, refletidas por caixas com bordas contínuas, que por seu turno são subdivididas em atividades ou ações, escritas por caixas com bordas tracejadas. Cada macro-atividade possui objetivos, responsáveis, precondições, entradas, atividades ou ações e saídas (MACIEL, 2022a), como pode ser identificado na Tabela 4.

Tabela 4 – Informação para cada atividade

Objetivos	$Objetivo_1, Objetivo_2...$
Responsáveis	$Pessoa_1, Pessoa_2...$
Precondições	$C_1, C_2...$
Entradas	$Entrada_1, Entrada_2...$
Atividades ou Ações	$Ação_1, Ação_2...$
Saídas	$Saída_1, Saída_2...$

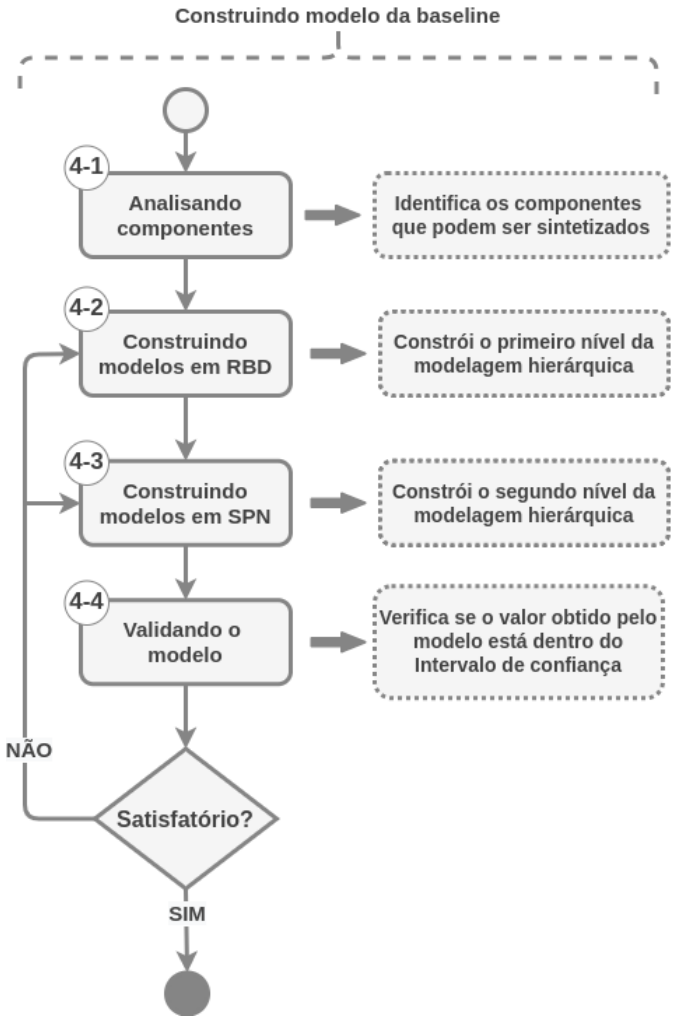
Figura 11 – Metodologia de avaliação



Os objetivos são estrategicamente traçados e segmentados considerando a natureza de cada atividade. A classificação do responsável é a pessoa que irá executar ou coordenar a execução das ações visando atingir o objetivo principal do processo. Em processos que executam várias atividades, podem existir mais de um responsável. As precondições podem variar em quantidade conforme a macro-tarefa, porém, sempre existirá como precondição, a finalização da macro-tarefa anterior. As entradas são os insumos do processo, tendo a possibilidade de variação da quantidade conforme a macro-tarefa. As ações ou atividades são representadas pelas caixas com bordas tracejadas e a sua quantidade também varia conforme o processo. Este conjunto de ações pode ser tão extenso que seja necessário a segmentação em várias atividades, podendo se transformar em outro fluxograma. A estratégia de segmentar as ações em outro fluxograma é utilizada na Atividade 4, onde ela se segmenta em outro fluxograma apresentado na Figura 12. Por último, as saídas são os entregáveis do processo, podendo ser listas, tabelas ou até mesmo

valores de métricas. Estas saídas geralmente serão utilizadas como entradas nos processos subsequentes. Na metodologia, só é possível o avanço para a macro-tarefa subsequente após a conclusão da atual.

Figura 12 – Metodologia de avaliação



No fluxograma existe ainda um losango, que possui a representação para uma etapa que pode levar a dois caminhos distintos. Essa decisão varia de acordo com o momento da metodologia. A primeira decisão é considerada satisfatória se o resultado alcançado pelo modelo construído estiver contido no intervalo de confiança. Caso seja satisfatória, pode-se dar seguimento ao fluxo transcorrendo para o passo seguinte. Caso contrário, retorna à tarefa de construção de modelos em RBD para realizar ajustes necessários. Este processo se repetirá quantas vezes que forem necessárias até ser atingido o objetivo de validação do modelo. A segunda e a terceira decisões são realizadas em conjunto com os administradores do sistema. Estas decisões analisam os resultados da disponibilidade encontrada no modelo visando identificar se o valor é considerado bom o suficiente na visão dos responsáveis pelo sistema. Os

caminhos a seguir dependerão da fase da metodologia, podendo passar para próxima fase ou retornar às fases anteriores. A metodologia só é finalizada no momento em que o administrador do sistema identificar que o ambiente proposto apresenta um valor de disponibilidade aceitável.

4.1.1 Avaliação da infraestrutura atual

A etapa de avaliação da infraestrutura atual representa as análises realizadas no ambiente estudado. Nesta fase da metodologia, não se sugere alterações de estrutura. Apenas é analisado o ambiente atual e criado um modelo com a sua representação. Como este trabalho analisa um sistema em produção, não é possível realizar injeções de falhas. Com isso, foi utilizada a abordagem do monitoramento para identificar pontos de falha e reparo do ambiente. A ferramenta utilizadas foi o Zabbix[1] e o período de monitoramento foi de seis meses. Neste período, a aplicação enfrentou vários momentos críticos em relação à quantidade de acessos como período de matrícula e inclusão de notas.

Nesta primeira etapa do estudo estão as primeiras quatro macro-tarefas da metodologia: estudando o sistema (1) ; monitorando o sistema (2); calculando os parâmetros (3); e construindo modelo da *baseline* (4). As fases estão detalhadas a seguir.

4.1.1.1 Macro-atividade 1: estudando o sistema

Este processo consiste em entender a estrutura do sistema. Quais são seus principais componentes físicos e lógicos, suas funcionalidades básicas, componentes existentes nas camadas físicas e virtuais, dentre outras características do ambiente. Os responsáveis são os administradores do sistema, que entendem a estrutura utilizada, juntamente com o autor deste trabalho. Por se tratar da primeira macro-atividade, a única precondição é que o ambiente esteja funcional, permitindo correta análise do ambiente. Como entrada para o processo estão as informações sobre os componentes, concebidas pelos administradores do sistema. Já as ações deste processo são a identificação da quantidade, da categoria e do relacionamento dos componentes. Foi identificado e classificado os componentes físicos em três camadas: conectividade, virtualização e armazenamento. Na camada de conectividade estão os *switchs ethernet*, na de virtualização estão os Servidores Físicos (SRV) e na de armazenamento estão os *switchs*

[1] https://www.zabbix.com

SAN e a unidade de armazenamento. Já nos componentes lógicos estão os Hipervisores (HP), SO das VMs, assim como as aplicações que funcionam dentro das VMs. É importante frisar que não foi considerado neste estudo, equipamentos dos sistemas de fornecimento energético nem de refrigeração. O entregável do processo é uma lista de componentes que precisam ser monitorados, possibilitando uma ação mais eficiente.

4.1.1.2 Macro-atividade 2: monitorando o sistema

Este processo tem o objetivo de realizar o monitoramento dos componentes do sistema visando a identificação das falhas e reparos. Para o presente estudo, essa atividade foi realizada pelo autor deste trabalho. Como precondição, está a necessidade da lista de componentes geradas na atividade anterior, bem como a classificação de todos os componentes do sistema. Tendo como insumo a lista da atividade anterior, foi configurado o monitoramento de cada componente, utilizando a ferramenta Zabbix. Foram realizados ajustes individuais para cada componente considerando a particularidade de cada um. Também determinou-se um prazo para coleta de dados sendo aguardado o tempo desejado para obter o cenário mais realista possível. O prazo determinado foi de seis meses. Durante este período, o sistema passou por várias etapas consideradas pelos seus administradores como críticas. São períodos onde o sistema precisa estar disponível e com capacidade adequada para atender aos clientes. Alguns destes períodos são: matrícula dos alunos, ajustes de matrícula e cadastro de notas. Nestes momentos uma indisponibilidade gera muitos problemas para o calendário acadêmico.

Foram realizados ajustes finos no monitoramento de cada componente virtual. Em alguns casos, foi configurado junto à equipe de desenvolvimento, consulta no banco de dados visando identificar as falhas. Como cada componente possui características distintas, não foi possível utilizar a mesma técnica de monitoramento para todos os componentes. Assim, para o SO e HP, foram configurados monitoramento de conectividade para identificar se estavam operacionais. Para as Camada do Balanceador de Carga (LB) e Camada de Aplicação (APP), foram utilizadas técnicas para conexões nos protocolos *Hypertext Transfer Protocol* (HTTP) e *Hypertext Transfer Protocol Secure* (HTTPS), porém, sempre respeitando as características individuais de cada camada. A individualidade do monitoramento também ocorreu para a Camada do Banco de Dados (BD), onde foram configuradas consultas diretas no Sistema Gerenciador de Banco de Dados (SGBD), retornando seu estado atual. Deste modo, se possibilitou um monitoramento diferenciado para cada camada da aplicação.

A coleta dos dados referentes aos componentes físicos foi realizada de maneira distinta. Alguns desses componentes são projetados visando diminuir ao máximo o tempo de indisponibilidade. Assim, naturalmente não apresentaram falhas no período de monitoramento. Desta forma, para os servidores físicos, *switches SAN*, *switches Ethernet* e unidade de armazenamento, os tempos de falha foram retirados da literatura. Por existir esta limitação do tempo de monitoramento, todos os dados poderiam ter sido retirados da literatura. No entanto, o ambiente estudado possui um contrato de manutenção diretamente com o fabricante destes equipamentos. O contrato de Acordo de Nível de Serviço (SLA) define que para troca de peças defeituosas o prazo máximo é de 48 horas para os *switches SAN* e unidade de armazenamento e 24 horas para os *switches ethernet* e servidor físico. Esses valores de SLA nos permite ter um tempo de reparo para cada equipamento. Desta forma, a combinação dos tempos retirados da literatura com os do SLA nos permite um cenário mais próximo à realidade do ambiente estudado.

Outra característica deste monitoramento é que para evitar uma quantidade demasiada de dados coletados, se identificou apenas os momentos de falhas e reparos de cada componente. Sempre que houve uma alteração no estado do componente, se coletou a data e hora para análise. Desta forma, foi obtido como resultado desta etapa uma lista com os tempos de falha e reparo de cada componente.

4.1.1.3 Macro-atividade 3: calculando os parâmetros

Nesta macro-atividade os valores dos parâmetros MTTR e MTTF de cada componente são calculados. Para o estudo apresentado, esta atividade foi realizada pelo autor deste trabalho. A precondição deste processo é a necessidade da existência dos valores referentes ao tempo de falha e reparo de cada componente. Para execução desta atividade, utiliza-se como insumo a lista gerada na tarefa anterior, que possui os registros de cada falha e reparo dos componentes. Nos momentos de falha e retorno ao correto funcionamento dos componentes, se guardou a data e hora da mudança de estado, conseguindo desta maneira identificar os períodos de indisponibilidade e atividade do componente, juntamente com a quantidade de vezes dos estados.

O cálculo do MTTF utiliza a Equação 2.9, passando o tempo total que o componente esteve no estado funcional e a quantidade de paradas. Para o cálculo do MTTR foi utilizado o mesmo raciocínio implementado pela Equação 2.10, passando o tempo total de indisponibilidade do

componente e a quantidade de paradas.

Como resultado desta atividade, foi gerada uma tabela com os MTTFs e MTTRs de cada componente. Esta atividade é muito importante, pois o resultado dos cálculos serão utilizados como insumo nos modelos construídos no passo seguinte.

4.1.1.4 Macro-atividade 4: construindo modelo da baseline

Este processo tem como objetivo principal construir um modelo que represente corretamente o ambiente estudado, classificado como *baseline*, permitindo análises mais precisas do ambiente e propostas de melhorias da disponibilidade. Para este estudo, essa macro-atividade foi realizada pelo autor deste trabalho, e as precondições para a realização são a existência dos valores dos MTTFs e MTTRs dos componentes e a relação existente entre eles. Este processo é bastante extenso, contemplando várias atividades. Desta forma, este processo foi segmentado em um sub fluxograma, permitindo melhor detalhamento das ações tomadas. Essas ações estão expostas no fluxograma da Figura 12, localizado na Página 58. A necessidade deste procedimento se concretizou pela importância e quantidade de ações realizadas. O resultado deste processo é um modelo hierárquico de dois níveis que se utiliza de modelos em RBD para calcular valores de MTTFs e MTTRs, e depois concatena suas saídas em um modelo de disponibilidade em SPN.

As especificações do sub fluxograma são as mesmas do fluxograma principal. Existem macro-tarefas e atividades, representadas por caixas com bordas contínuas e tracejadas, respectivamente. O losango representa a possibilidade de dois caminhos distintos. Caso a decisão seja positiva o fluxo continua, caso contrário, volta a uma etapa devidamente predefinida.

Existe uma particularidade nesta etapa. Por ser subdividida, ela finaliza a sua subdivisão após um questionamento. Caso o resultado deste questionamento seja positivo, a etapa finaliza a subdivisão e já é redirecionada para outro questionamento, já no fluxograma principal, que, por seu turno, questiona se o valor da disponibilidade obtido no sub fluxograma é considerado satisfatório como resultado do modelo. Se o valor da disponibilidade alcançada for satisfatório para os administradores do sistema, não precisa realizar melhorias no ambiente e todo o processo é finalizado. Caso não seja, se inicia a etapa de planejamento de estruturas alternativas.

Macro-atividade 4.1 - Analisando componentes:

O primeiro processo do sub fluxograma tem como objetivo analisar as relações entre os componentes em busca da identificação de quais componentes podem ser sintetizados em modelos RBD. Um segundo objetivo é identificar a relação entre componentes e serviços, e identificar quais componentes possuem redundância. Para o estudo realizado, essa macro-atividade foi realizada pelo autor deste trabalho. Este processo se utiliza da lista dos valores dos MTTFs e MTTRs dos componentes e a relação existente entre eles. A principal ação é a análise minuciosa dos componentes, identificando qual nível de redundância nas camadas da aplicação. Foi identificada a existência de componentes que possuem redundância e operam em HA. Já outros não possuem redundâncias, porém, se correlacionam uns com os outros. Esta relação pode ocorrer pela conectividade física ou lógica entre eles. Independente da relação nestes casos, se um componente apresentar defeito, mesmo que o outro esteja no estado funcional, o serviço ofertado por eles apresentará indisponibilidade. Um exemplo é a disponibilidade do Serviço do LB. Este serviço é composto pelo SO do balanceador (SO_{ba}) e pelo serviço do balanceador (SER_{ba}). Caso o SO_{ba} apresente defeito, consequentemente o SER_{ba} também apresentará defeito e o LB estará indisponível. Já no sentido contrário, caso o SER_{ba} apresente defeito, o SO_{ba} pode estar operacional, mas o LB estará indisponível da mesma forma. Este tipo de interação caracteriza uma relação em série destes componentes. Desta forma, caso um componente apresente defeito, todo serviço entra no estado de indisponibilidade. Como saída deste processo, é criada uma lista de componentes, identificando a relação de redundância existente entre eles.

Macro-atividade 4.2 - Construindo modelos em RBD:

Esta macro-atividade representa o primeiro nível da modelagem hierárquica. O objetivo principal é a construção de modelos em RBD. Para presente estudo, essa atividade foi realizada pelo autor deste trabalho, tendo como precondições e entrada, a lista obtida no processo anterior. As atividades deste processo são criações de modelos em RBD em série, que representam as camadas do sistema onde os componentes possuem uma conexão entre si e todos precisam estar no estado operacional para que o sistema esteja funcionando corretamente. O ambiente estudado possui muitos componentes tornando a modelagem em SPN custosa computacionalmente. Esta situação ocorre pela característica da SPN que, neste caso, cria uma matriz com muitos estados. Uma estratégia para amenizar essa limitação computacional

é, sempre que possível, associar componentes em um único serviço. Para isto, foi adotada a estratégia de utilizar modelos em RBD. Os valores de MTTF e MTTR resultantes do modelo RBD serão utilizados na próxima etapa como insumo para o modelo em SPN. Entretanto, nem todos os serviços podem ser simplificados. A simplificação dependerá da configuração de cada serviço.

Todos os valores obtidos no estudo são exponencialmente distribuídos. Os serviços que possuem redundância utilizando HA, precisam ser modelados em RBD utilizando composição paralela. Entretanto, na modelagem em RBD, ao sintetizar componentes em paralelo que possuam tempos exponencialmente distribuídos, o resultado da sintetização não é exponencialmente distribuído. Desta forma, não foi possível implementar esta sintetização nas transições do modelo SPN. Por outro lado, os modelos em RBD que utilizam componentes em série, após a sintetização, o resultado continua sendo exponencialmente distribuído, podendo ser utilizado no modelo SPN. Portanto, apenas foi sintetizado os serviços que possuem componentes operando em série.

No ambiente estudado, foram sintetizados os seguintes componentes: servidor, utilizando dados sobre o servidor físico e o hipervisor balanceador de carga, utilizando os dados do SO e o serviço do baleador; a aplicação, utilizando dados do SO e serviço da aplicação; e por último o banco de dados, utilizando os dados do SO e o serviço do banco de dados.

Como saída para este processo estão os tempos de MTTFs e MTTRs de cada camada sintetizada.

Macro-atividade 4.3 - Construindo modelos em SPN:

O processo em questão representa o segundo nível da modelagem hierárquica, tendo seu principal objetivo a concepção do modelo de disponibilidade em SPN que representa a *baseline*. Assim como em outras etapas, essa atividade foi realizada pelo autor deste trabalho. A precondição é a existência de valores para os parâmetros MTTF e MTTR de cada componente que será modelo, assim como o nível de redundância de cada camada. Como entrada para o modelo, se utiliza os dados sintetizados na atividade anterior e os colhidos através da etapa de monitoramento do fluxograma principal. O entregável desta macro-atividade é a apresentação do modelo de disponibilidade em SPN, que represente a *baseline*, em conjunto com o valor da disponibilidade encontrada.

Macro-atividade 4.4 - Validando o modelo:

Este processo difere um pouco dos demais apresentados no sub fluxograma. O seu objetivo principal é a validação do modelo criado na macro-atividade anterior. Para o estudo realizado, essa macro-atividade tem como responsável o autor deste trabalho e os administradores do sistema. A precondição para execução deste processo é a existência do modelo que represente o ambiente estudado. Como entrada para a validação, se utiliza o valor da disponibilidade encontrada no modelo anterior, o valor da disponibilidade encontrada através do monitoramento e o intervalo de confiança.

Para haver uma comparação matematicamente aceitável, é necessário a existência de um intervalo de confiança que permita a comparação do valor obtido no modelo com os valores reais do ambiente. As ações do processo são a criação do intervalo de confiança, o cálculo da disponibilidade com dados obtidos no monitoramento, a comparação dos valores da disponibilidade e identificação se os valores estão satisfatórios.

No cálculo da disponibilidade obtido através do monitoramento, se utilizou as equações e valores que serão apresentadas no Capítulo 7, na Subseção 7.1.3, na Página 106. Na validação, aplicou-se como estratégia a criação de um intervalo de confiança de 95%, utilizando a técnica de *bootstrap*. Se o valor obtido na etapa anterior estiver contido neste intervalo, não se pode refutar que o modelo não representa a realidade e o sub fluxograma é finalizado. Caso o valor não esteja contido no intervalo, analisam-se os erros e se retrocede para a Atividade 4.2 ou 4.3 para reconstruir o modelo. Este processo pode ser repetido quantas vezes for necessário até um resultado positivo.

4.1.2 Planejamento de estruturas alternativas

A etapa de planejamento de estruturas alternativas representa na metodologia o desenvolvimento das propostas de infraestrutura. As atividades desta etapa da metodologia se repetem para cada infraestrutura proposta, tendo como objetivo principal a concepção de um modelo de disponibilidade em SPN que apresente uma melhoria na disponibilidade do sistema estudado.

A primeira macro-atividades desta etapa é uma análise de sensibilidade (5), seguida da construção do modelo evolutivo (6), análise e apresentação dos resultados (7 e 8), respectivamente.

4.1.2.1 Macro-atividade 5: conduzindo a análise de sensibilidade

O processo da análise de sensibilidade tem como objetivo identificar os componentes que podem ser ajustados, tendendo a incrementação da disponibilidade do ambiente. Esta atividade visa ajustar o ambiente com o menor número de alterações possíveis, permitindo alterações mais assertivas. Para o estudo realizado, essa macro-atividade foi realizada pelo autor deste trabalho e a precondição para a sua realização é a existência de um modelo em SPN. A entrada do processo será sempre os valores de MTTF e MTTR dos componentes do modelo que se deseja analisar.

A análise de sensibilidade de um modelo é composta por um conjunto de ações que devem ser realizadas. Para cada componente, são configuradas alterações dos valores do MTTF e MTTR, porém, estas alterações precisam ser realizadas individualmente. Visando manter um equilíbrio nas alterações dos valores, sem permitir tendências nas alterações, foram configurados ajustes nos valores dos parâmetros de 50% para mais e 50% para menos, com intervalos de 10% entre as alterações, resultando no total de 10 alterações para cada parâmetro. Após cada alteração se analisa o impacto na métrica da disponibilidade do ambiente como um todo. Este procedimento é realizado para cada parâmetro de cada componente. No final das 10 alterações do parâmetro, se utiliza a Equação 2.15 para se identificar o índice de sensibilidade daquele parâmetro. Estas ações são repetidas para todos os componentes do ambiente. A saída do processo é uma lista com o índice de sensibilidade de todos os componentes, tendo cada um deles dois parâmetros, MTTF e MTTR. Quanto maior o índice, mais significativa é o impacto deste parâmetro na disponibilidade final do ambiente.

4.1.2.2 Macro-atividade 6: construindo modelo evolutivo

Esta macro-atividade tem como principal objetivo a construção de um modelo evolutivo de disponibilidade para o ambiente estudado. A atividade de estudo foi realizada pelo autor deste trabalho. A construção deste modelo está vinculada com a necessidade da lista de sensibilidade, tendo como entrada um ou mais componentes que serão aprimorados. Existem várias técnicas que possibilitam o aprimoramento da disponibilidade, porém, todas possuem o propósito de aumentar o valor do MTTF, deixando o componente mais tempo disponível e/ou diminuir o MTTR, permitindo que o componente passe menos tempo indisponível. A técnica utilizada neste estudo foi a redundância dos componentes. Inicialmente, se aplicou duas formas de

redundância, *cold standby* e HA. Foram escolhidos essas formas de redundâncias por serem as configurações utilizadas no ambiente de produção do sistema estudado.

Os primeiros parâmetros a serem ajustados são os que possuem o maior índice de sensibilidade por possibilitarem um maior impacto na disponibilidade do ambiente. Assim, pode-se conseguir resultados mais assertivos com um número menor de modificações do ambiente. Após a escolha do parâmetro foram aplicadas as devidas redundâncias no modelo SPN e calculada a disponibilidade do ambiente. O entregável desta atividade é um novo modelo de disponibilidade em SPN referenciando um ambiente proposto com sugestões na infraestrutura.

4.1.2.3 Macro-atividade 7: analisando os resultados

Este processo é puramente administrativo e a próxima fase sempre será um questionamento. A etapa em questão não possui intervenções nos modelos ou sugestões de mudanças, utilizando apenas o resultado obtido no modelo anterior para analisar a métrica de disponibilidade e seu respectivo tempo de indisponibilidade. Esta análise é realiza em conjunto com os administradores do sistema e o autor deste trabalho. A precondição é a existência de um modelo e valores obtidos neste modelo para análise dos resultados. Pelo fato deste trabalho ser voltado para melhora da disponibilidade do ambiente, se utiliza esta métrica como base à ser analisada.

Caso o resultado desta ação seja considerado satisfatório, pode-se passar para a próxima etapa de apresentação dos resultados. Caso contrário, volta-se para a etapa anterior para aprimorar o modelo. Este processo se repete quantas vezes forem necessárias até conseguir um valor aceitável para a disponibilidade do ambiente.

4.1.2.4 Macro-atividade 8: apresentando os resultados

Neste processo são apresentadas as melhorias da disponibilidade obtidas nos ambientes propostos. A exposição dos resultados, bem como os estudos aplicados, foram realizados pelo autor deste trabalho. As precondições e as entradas são os valores obtidos nos estudos de casos. A ação é a elaboração visual dos resultados. Os entregáveis desta macro-tarefa são os gráficos de comparação de disponibilidade e tempo de indisponibilidade dos estudos de casos, apresentação da análise de sensibilidade e exposição dos modelos propostos para a melhoria da disponibilidade do sistema.

5 ARQUITETURA DO SISTEMA

Este trabalho analisa a disponibilidade e propõe mudanças na infraestrutura de um sistema responsável pela gerência acadêmica dos alunos de uma universidade brasileira de grande porte, visando melhoria do índice analisado. Para atingir este objetivo foi necessário identificar, analisar e propor alterações nos componentes físicos e lógicos do sistema. A análise foi iniciada nos componentes físicos do CD e finalizada nos componentes lógicos da aplicação. Cada componente possui sua particularidade e importância no ambiente.

O sistema estudado encontra-se hospedado em um CD próprio da universidade e não possui interações com nuvens públicas. Todavia, utiliza estrutura de uma nuvem privada. A abordagem inicial foi a identificação de todos os componentes que fazem parte do CD. Foram identificados e segmentados os componentes físicos do CD em três grupos de infraestrutura: energia, refrigeração e TI. Este estudo analisa e propõe alterações apenas na infraestrutura de TI. Não foi analisado detalhadamente os demais grupos, embora se tenha ciência da importância de ambos na disponibilidade do complexo sistema de TI. Propor alterações nos grupos de energia e refrigeração neste estudo abriria vastas áreas de discussões, possibilitando o distanciamento do nosso objetivo principal que é identificar os pontos sensíveis da infraestrutura de TI e o impacto de uma redundância destes pontos na disponibilidade do sistema, incluindo redundância do CD.

Outra análise que não foi trazida para o estudo foi a modularização da aplicação, embora se conheça os impactos positivos da arquitetura modular na disponibilidade e flexibilização do ambiente. Este tópico não será abordado neste estudo, uma vez que não há tratativa sobre desenvolvimento de sistemas. O foco do estudo está na estrutura e quantidades de camadas virtuais e recursos de infraestrutura física como servidores e CD.

Este sistema possui uma estrutura bastante utilizada em computação em nuvem. Por este motivo, o presente estudo pode ser aplicado em qualquer sistema que possua estruturas semelhantes ou que possam ser minimamente adaptáveis. Para melhor compreensão, as análises foram segmentadas em três grupos de estrutura: centro de dados, computacional e lógica. A estrutura de centro de dados aglutina os grupos de refrigeração e energia, exposto detalhadamente na Seção 5.1. A estrutura computacional se refere à processamento, conectividade e armazenamento e está detalhada na Seção 5.2. Por último, a Seção 5.3 detalha todo o funcionamento e segmentação lógica do sistema (CLEMENTE et al., 2022).

5.1 ESTRUTURA DO CENTRO DE DADOS

O primeiro grupo a ser detalhado é a Estrutura do Centro de Dados (DCS). Este grupo representa os componentes responsáveis pela refrigeração e distribuição de energia do CD. A configuração atual da DCS está representada pela Figura 13. Nela, pode-se identificar componentes específicos e fazer uma relação com o modelo de classificação utilizada pela ANSI/TIA-942-A, explicada na Seção 2.1. A DCS estudada não possui uma certificação de classificação, porém, a sua configuração pode ser comparada com um CD *Tier* II.

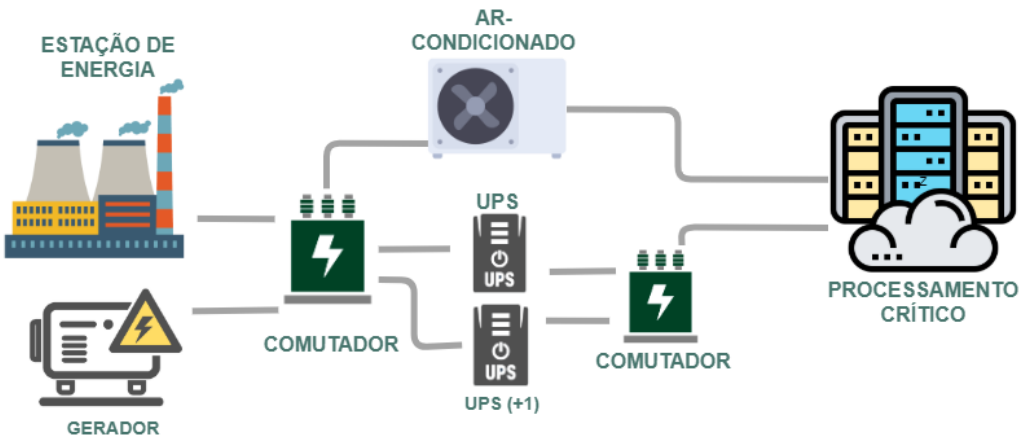


Figura 13 – Estrutura do Centro de Dados (DCS).

A DCS possui uma fonte de energia externa, denominada no trabalho como **estação de energia**. Esta configuração não possui redundância, tendo a sua classificação como **N**. Também não possui mais de um caminho para fornecimento externo de energia. A autonomia deste componente pode ser considerada contínua por não existir limite de tempo para o fornecimento energético. Logicamente o fornecimento energético será interrompido caso ocorra problemas oriundos da estação de energia, entrando em funcionamento o sistema de geração de energia interno.

A fonte de energia interna possui dois componentes, o gerador e a UPS. O **gerador** é responsável pelo fornecimento energético caso ocorra falhas no fornecimento externo de energia. Esta configuração não possui redundância e sua classificação é **N**. A potência energética gerada pelo equipamento é suficiente para o abastecimento do CD e possui uma autonomia de seis horas ininterruptas. Este tempo de autonomia pode ser aumentado caso haja abastecimento do equipamento, procedimento que pode ser realizado com o equipamento em funcionamento.

O segundo componente do grupo é a **UPS**, que possui as funções de estabilizar a corrente elétrica e atuar como fonte de alimentação ininterrupta. A UPS utiliza baterias para

fornecimento energético, que devem estar previamente carregadas. Este sistema entra em funcionamento sempre que ocorre interrupção no fornecimento de energia, tanto da estação de energia como do gerador. Na estrutura estudada, existem duas UPS funcionando em alta disponibilidade. Desta maneira, existem dois caminhos de conexões energéticas com o CD e sua classificação de **2N**. Caso uma UPS apresente falhas, a outra pode atuar normalmente. A autonomia do conjunto de UPS é de quatro horas de fornecimento elétrico para a carga utilizada no CD.

Outro sistema existente na DCS é o **sistema de refrigeração**, que consiste em oito dispositivos individuais de resfriamento operando em alta disponibilidade, tendo todos os equipamentos em funcionamento simultâneo. Este sistema não possui autonomia energética. Caso ocorra falha na distribuição elétrica, automaticamente os dispositivos se tornam inoperantes. O sistema de refrigeração é alimentado energeticamente pela estação de energia e gerador. Não recebe carga elétrica das UPS.

A configuração atual da DCS permite autonomia energética de sete horas para o CD, caso ocorra falha no abastecimento energético da estação de energia. Este tempo considera o não abastecimento do gerador. Caso o abastecimento ocorra, o tempo de autonomia existirá enquanto houver combustível ou o gerador estiver em correto funcionamento. Após a falta de abastecimento energético oriundo do gerador, as UPS continuam com o fornecimento de energia para os equipamentos dentro do CD. A autonomia das UPS para a carga atual de equipamentos são de quatro horas de funcionamento. Todavia, este tempo não pode ser alcançado por deficiência de refrigeração, tendo em vista que o sistema de refrigeração não recebe energia das UPS. Após o desligamento do gerador, os equipamentos de refrigeração também são desligados.

Em resumo, caso ocorra uma falha energética externa e uma falha no gerador, o tempo máximo de funcionamento do CD é de uma hora. Após este tempo, mesmo existindo energia oriunda da UPS, os equipamentos precisam ser desligados para evitar danos por superaquecimento. Caso ocorra uma falha externa no fornecimento energético e o gerador funcione corretamente, até acabar o combustível, o tempo de autonomia energética do CD é de aproximadamente sete horas.

5.2 ESTRUTURA COMPUTACIONAL

A estrutura computacional é composta pelos componentes físicos da infraestrutura de TI. Esta segmentação está apresentada na Figura 14. Para melhor entendimento, a estrutura foi dividida em três camadas utilizando como parâmetro as funcionalidades dos equipamentos, sendo elas: a camada de conectividade, a camada de virtualização e a camada de armazenamento.

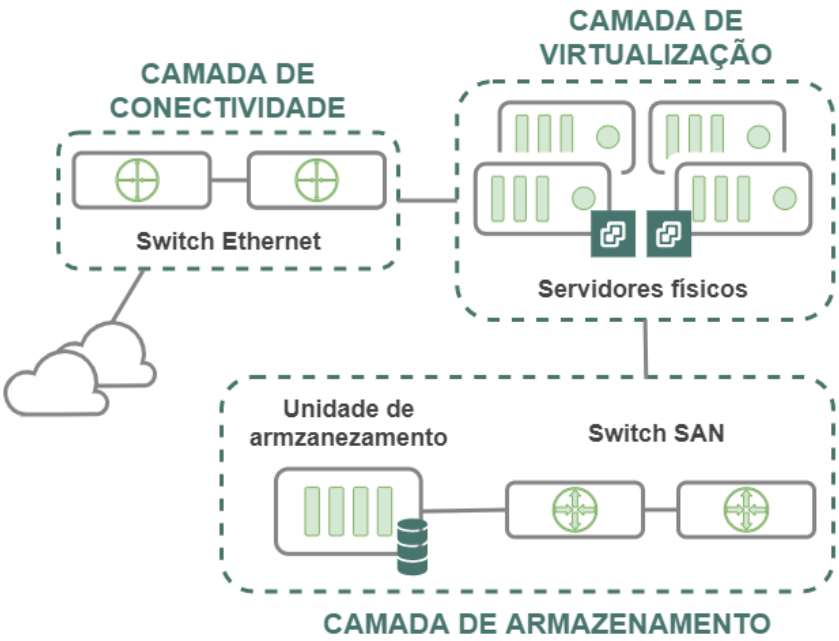


Figura 14 – Estrutura lógica

A primeira camada a ser apresentada é a **camada de conectividade** que é composta por dois *Switchs Core Ethernet* (NET) do modelo *BlackDiamond 8800 Series Switches*[1] operando em alta disponibilidade, utilizando o modo ativo/ativo. Esta camada é a responsável por ofertar o serviço à comunidade externa.

A segunda camada é a **camada de virtualização**. Visando a facilidade no entendimento e na modelagem do ambiente, se aglutinou os SRV, com o SO instalado neles. Neste estudo, foi utilizado um sistema operacional de virtualização de propriedade da VMware[2], conhecido como ESXi[3], na versão 6.5. Os servidores físicos possuem o modelo *HP ProLiant DL380p Gen8 Server*[4], tendo sua quantidade e redundância variando conforme o momento do estudo.

[1] www.extremenetworks-ua.com
[2] www.vmware.com
[3] www.vmware.com/products/esxi-and-esx.html
[4] www.hpe.com

Na estrutura inicial (*baseline*), existem três servidores, porém não funcionam na configuração de redundância.

Os recursos físicos dos servidores também são variáveis e dependerá da funcionalidade de cada um. Todos os servidores possuem a mesma capacidade de processamento, diferenciando entre eles a quantidade de RAM. São três configurações possíveis:

- *Servidor de pequeno porte:* servidor com dois processadores dual-core e com 12 GB de RAM, utilizado para hospedar as camadas de orquestração e balanceador de carga.
- *Servidor de médio porte:* servidor com dois processadores dual-core e com 24 GB de RAM, utilizado para hospedar a camada de aplicação.
- *Servidor de grande porte:* servidor com dois processadores dual-core e com 32 GB de memória RAM, utilizado para hospedar a camada de banco de dados.

A última camada a ser apresentada é a **camada de armazenamento**. Nela, existem duas categorias de equipamentos, os **SAN** e a **Unidade de Armazenamento (STG)**. Os switches SAN são do modelo *Switch Fibre Channel HPE SN6700B série B*[5] e estão configurados para operar em alta disponibilidade. Estes componentes são responsáveis pela conexão dos servidores físicos com a unidade de armazenamento. Já a STG é responsável pelo armazenamento de todas as VMs. Existe apenas uma STG do modelo *HP 3PAR StoreServ 7400 Storage*[6], porém, a sua disponibilidade é altíssima por ser um equipamento voltado para ambientes robustos. Os seus componentes internos possuem diversos recursos de redundância (física e lógica).

A camada de conectividade é diretamente responsável pela disponibilidade do ambiente estudado. Por possibilitar o acesso externo ao ambiente, caso apresente falha, o sistema se torna indisponível mesmo que internamente todas as aplicações estejam no estado funcional. As camadas de virtualização e armazenamento também interferem diretamente na disponibilidade do sistema, porém de uma forma muito mais incisiva. Por ser responsável pelo processamento, caso haja problemas na camada de virtualização, as VMs hospedadas no servidor se desligarão imediatamente. Situação semelhante ocorre ao apresentar indisponibilidade na camada de armazenamento, porém, por ser responsável pelo armazenamento de todas as máquinas virtuais, ao apresentar defeito, todo o ambiente se torna indisponível imediatamente. Outra possibilidade de falha na camada de armazenamento acontece caso haja problemas na conexão

[5] www.hpe.com/psnow/doc/a50002550enw?section=Product%20Documentation
[6] support.hpe.com/hpesc/public/docDisplay?docId=c03560484

de apenas um servidor à unidade de armazenamento. Neste caso, apenas as máquinas virtuais hospedadas naquele servidor ficarão indisponíveis. Caso o servidor possua alguma categoria de redundância, as máquinas virtuais serão iniciadas em outro servidor, podendo ou não afetar na disponibilidade do sistema estudado.

5.3 ESTRUTURA LÓGICA

A estrutura lógica compõe os componentes virtuais da infraestrutura de TI. Ela representa os *softwares* responsáveis pelo funcionamento do sistema, denominada *camada de software*. O sistema possui uma estrutura que nos permite agrupar componentes com as mesmas funcionalidades. Desta forma, foi possível segmentar a *camada de software* em quatro subcamadas: **Camada de Orquestração (VC), LB, APP, BD**.

As subcamadas são compostas por VMs hospedadas na camada de virtualização. Cada subcamada possui um SO (CentOS Linux) e a aplicação responsável por um serviço específico, conforme mostrado na Figura 15.

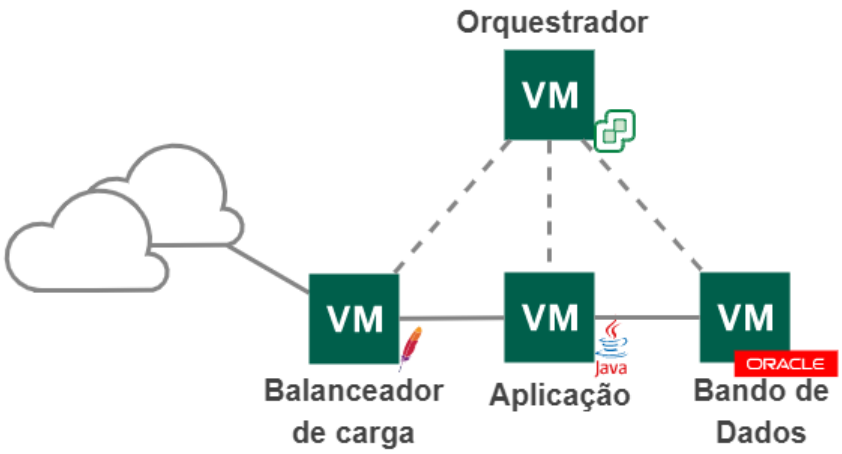


Figura 15 – Estrutura lógica.

A primeira camada da estrutura lógica a ser apresentada é a **Camada de Orquestração (VC)**, que possui a responsabilidade de manter a alta disponibilidade do ambiente. Ela executa o monitoramento das demais subcamadas em busca de componentes que estejam no estado de falha e tenta deixá-los disponíveis novamente. Por esta característica, ela não faz parte diretamente da estrutura do sistema, pois não recebe nenhuma requisição de acesso.

O sistema utilizado para esta funcionalidade é o VCenter[7], responsável por gerenciar os recursos físicos do conjunto de servidores, o desempenho do ambiente, a disponibilidade das

[7] www.vmware.com/br/products/vcenter-server.html

máquinas virtuais, entre outras funções. O VCenter usa um recurso conhecido como HA que permite a identificação de mau funcionamento do servidor e executa a migração automática das VMs hospedadas nele para outro servidor operacional. Nesse processo, o VCenter identifica qual servidor tem a maior disponibilidade de recursos e em seguida executa a inicialização da VM no servidor escolhido. Mesmo com a capacidade de identificar qual servidor possui os melhores recursos físicos, o HA é uma ação reativa, pois só é realizada após a falha do servidor. Assim, a VM também sofre uma interrupção na prestação de serviços e só após essa pequena interrupção o VCenter pode ligá-la novamente em outro servidor.

Outra funcionalidade utilizada no ambiente é o *Distributed Resource Scheduler* (DRS). Ela consiste em identificar e distribuir a carga de trabalho das VMs pelos servidores previamente configurados. Podem ser configuradas regras de afinidade, onde se permite ou não que a VM funcione em determinados servidores. O VCenter possui vários outros recursos, que não serão abordados neste estudo. Este conjunto de habilidades classifica o VCenter como um orquestrador de ambiente virtual. A camada de orquestração é hospedada em um servidor de pequeno porte e possui seis processadores virtuais e seis GB de RAM. A versão do sistema de orquestração é a 6.7.

A **Camada de Balanceador de Carga (LB)** é a primeira camada do sistema, responsável pela interação da aplicação com o cliente. Esta camada recebe as solicitações dos clientes e as encaminha para a camada de aplicação. Uma de suas funções é identificar qual componente da camada de aplicação está com a maior disponibilidade de recursos e redirecionar as solicitações para ele. Esta funcionalidade é conhecida como balanceamento de carga. O balanceador de carga usado nesta solução é um sistema gratuito conhecido como Apache[8], na versão 2.4.6. A camada de balanceador de carga também está hospedada em um servidor de pequeno porte e possui dois processadores virtuais e dois GB de RAM.

A **Camada de Aplicação (APP)** é a segunda camada do sistema, sendo a aplicação propriamente dita. Esta camada é responsável por toda lógica de negócios e processamento de informações. A linguagem utilizada é Java[9], versão gratuita de propriedade da Oracle. A aplicação foi construída utilizando a versão do java 1.6.0.45, atualmente defasada, e de forma monolítica. Como já foi dito, não será proposto alterações na estrutura de construção da aplicação. É utilizada a plataforma de contêiner gratuita conhecida como Apache Tomcat[10],

[8] https://www.apache.org
[9] https://www.oracle.com/br/java
[10] https://tomcat.apache.org

na versão 6.032 para armazenar a aplicação. A camada de aplicação é hospedada em um servidor de médio porte e possui quatro processadores virtuais e quatro GB de RAM.

A terceira e última camada do sistema é a **Camada de Banco de Dados (BD)**. Esta camada é responsável pelo armazenamento dos dados inseridos no sistema, e utiliza uma aplicação proprietária da *Oracle Enterprise*[11], na versão 10g. Está hospedada em um servidor de grande porte, que possui oito processadores virtuais e 30 GB de memória RAM.

5.4 CONSIDERAÇÕES FINAIS

As VMs são guardadas na camada de armazenamento e podem ser acessadas por todos os servidores. Em casos de falha de um destes servidores, as VMs podem ser iniciadas (manualmente ou automaticamente) em qualquer servidor conectado e previamente definido. Uma decisão adotada no estudo foi não reunir VMs de subcamadas diferentes no mesmo servidor, exceto para a Camada de Orquestração e a Camada do Balanceador de Carga. Outro cuidado adotado foi de amenizar o desperdício de recursos. Para insto, se decidiu usar a capacidade total dos servidores físicos sempre hospedando o número máximo de VMs suportadas. Desta forma, uma configuração de servidor de pequeno porte pode hospedar no máximo um orquestrador e dois balanceadores de carga. Uma configuração de servidor de médio porte pode hospedar no máximo cinco aplicações e uma configuração de servidor de grande porte pode hospedar no máximo um banco de dados.

[11] https://www.oracle.com

6 MODELO DE DISPONIBILIDADE

A proposta principal do trabalho é propor modelos possíveis de serem executados com os recursos físicos do ambiente estudado, e que possibilitem uma melhoria na disponibilidade do sistema estudado. Para ajudar com este objetivo, é utilizada a estratégia de concepção de modelos para representar o ambiente e, assim, elaborar os cenários. Esta estratégia é amplamente utilizada nos estudos de TI, por facilitar uma visualização global do ambiente (MARSAN; CONTE; BALBO, 1984; MOLLOY, 1981; MACIEL, 2016; MACIEL et al., 2012a). Utilizando modelos, consegue-se identificar a quantidade de componentes, as relações entre eles, realizar análises e cálculos referentes a várias medidas.

Este capítulo apresenta os modelos de disponibilidade que representam a configuração atual do ambiente estudado e os cenários propostos utilizados no estudo. Visando facilitar o entendimento, está sendo apresentado os modelos separados por estruturas (computacional, lógica e do CD). A ferramenta Mercury[1] é utilizada (PINHEIRO et al., 2021) para construir os modelos, realizar as análises de sensibilidade e medidas de disponibilidade.

Uma modelagem de disponibilidade hierárquica usando modelos em RBD e SPN é proposta. Essa abordagem é benéfica para analisar sistemas em nuvem (MATOS et al., 2015). Utilizando este artifício, é possível combinar componentes estratégicos do sistema, facilitando a identificação e realização dos cálculos. Além disso, o SPN permite incluir ações temporizadas, simplificação de componentes, representações de ações de alta disponibilidade como migrações de VMs ligadas ou desligadas. Em outras palavras, modelos de alto nível mais fiéis ao cenário estudado podem ser propostos. A Figura 16 representa o diagrama hierárquico utilizado neste estudo.

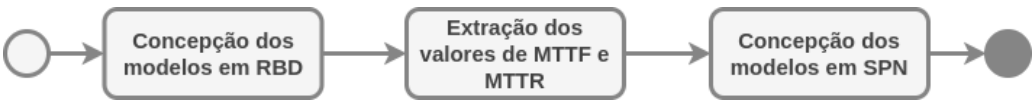


Figura 16 – Diagrama da modelagem hierárquica.

A modelagem hierárquica aqui apresentada possui seu primeiro nível nos modelos em RBD e o segundo nível são os modelos em SPN. Os resultados dos modelos em RBD são utilizados como insumos para a modelagem em SPN. Todo o ambiente poderia ser modelado diretamente em SPN, porém, por possuir muitos componentes torna a modelagem em SPN custosa computacionalmente. Isto ocorre, porque a modelagem em SPN produz uma matriz

[1] http://www.modcs.org/

com muitos estados. Uma estratégia para amenizar essa dependência computacional é, sempre que possível, simplificar componentes em um único serviço. Para isto, foi adotada a utilização de modelos em RBD. Os valores de MTTF e MTTR resultantes do modelo RBD são utilizados como insumo para o modelo em SPN.

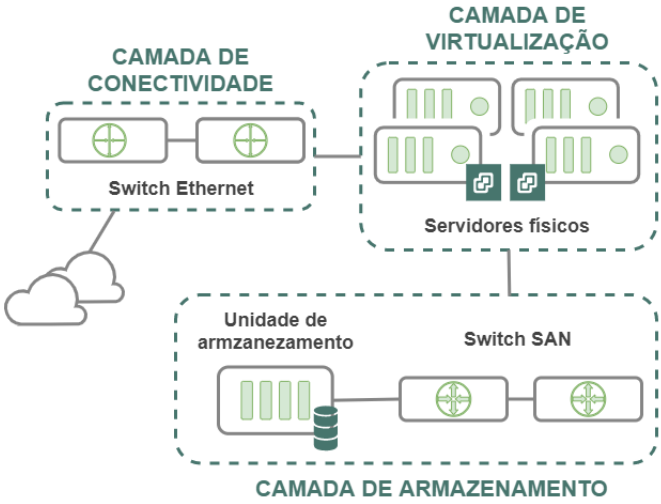
Entretanto, nem todos os serviços podem ser simplificados. Isto dependerá da configuração e característica de cada serviço. A modelagem em paralelo no RBD, ao utilizar componentes com tempos exponencialmente distribuídos para o MTTF e MTTR, não gera um resultado exponencialmente distribuído. Uma das funções dos modelos em RBD é gerar tempos de MTTF e MTTR exponencialmente distribuídos para serem aplicados nas transições temporizadas do SPN. Assim, no ambiente estudado, os componentes configurados em HA não foram simplificados. Outra característica do RBD que impediu a sua utilização para simplificação de componentes, foi que ao ser utilizado para descoberta dos tempos de MTTF e MTTR, realiza o cálculo de confiabilidade, resultando em valores não reais para o cenário desejado. Isso ocorre, pois, nos modelos de confiabilidade, ao identificar um defeito em algum componente, não considera a possibilidade de reparo. Esta característica não corresponde com a realidade do estudo, já que existe o SLA dos equipamentos para reposição de peças e equipes de reparo para os sistemas virtuais.

Por outro lado, na modelagem em série no RBD, quando utilizados componentes com tempos exponencialmente distribuídos, o resultado continua sendo exponencialmente distribuído, podendo ser utilizado nas transições temporizadas no modelo SPN. Portanto, neste trabalho, é adotada a simplificação de componentes para serviços que possuam componentes operando em série. Desta forma, apenas as camadas de virtualização, orquestração, balanceamento de carga, aplicação e banco de dados passaram pela modelagem hierárquica.

Este capítulo segmenta as modelagens em seções. A Seção 6.1 representa a modelagem da estrutura computacional, onde os modelos utilizados para as camadas de conectividade, virtualização e armazenamento são expostos. Em seguida, os modelos que representam a estrutura lógica do sistema estudado na Seção 6.2 são apresentados. Essa seção apresenta os dois níveis de modelagem hierárquica que representam as camadas de orquestração, balanceamento de carga, aplicação e banco de dados. A Seção 6.3 se refere à estrutura do CD. Nessa seção, foram utilizadas sintetizações das estruturas lógicas e computacional visando a representação do CD atual. Na Seção 6.4, um modelo referente à redundância de CD é mostrado, contemplando uma estrutura de controle que permite identificar o CD ativo.

6.1 ESTRUTURA COMPUTACIONAL

Nesta seção são apresentados os modelos hierárquicos utilizados na representação da estrutura computacional. Para facilitar o entendimento, a Figura 14, que representa a estrutura computacional, exposta previamente na Seção 5.2 é reapresentada.



A estrutura computacional corresponde aos componentes físicos da infraestrutura de TI, composta pelas camadas de conectividade, virtualização e armazenamento. A modelagem destas camadas estão expostas separadamente nas Subseções 6.1.1, 6.1.2 e 6.1.3, respectivamente. O passo inicial para a elaboração destes modelos é a identificação da quantidade de componentes e suas redundâncias. Esta atividade foi realizada e explicada na Metodologia 4.1, mais especificamente na Macro-atividade 4.1. Como resultado desta macro-atividade, se obteve as informações referentes a redundância dos equipamentos. A Tabela 5 representa esta informação.

Tabela 5 – Relação dos componentes da estrutura computacional.

Camada	**Componente(s)**	**Tipo de redundância**
Conectividade	Switch Ethernet	HA
Virtualização	Servidor físico (HW) Hipervisores (HP)	Série
Armazenamento	Unidade de armazenamento Switch SAN	HA

Os componentes que operam em HA foram modelados diretamente em SPN, pois os tempos exponencialmente distribuídos podem ser aplicados diretamente nas transições tem-

porizadas. Em contrapartida, os componentes que atuam em série foram modelados em RBD pelos motivos previamente apresentados na introdução do Capítulo (6).

6.1.1 Camada de conectividade

A modelagem da camada de conectividade está exposta nesta seção. Como explicado na Seção 5.2, ela é a responsável por possibilitar o acesso externo ao sistema estudado. Possui dois equipamentos operando em HA, e teve sua representação modelada diretamente em SPN. Na modelagem, utilizando a estrutura básica, apresentada no Capítulo 2.5.2 na Página 41. O modelo representando a camada de conectividade está apresentada na Figura 17.

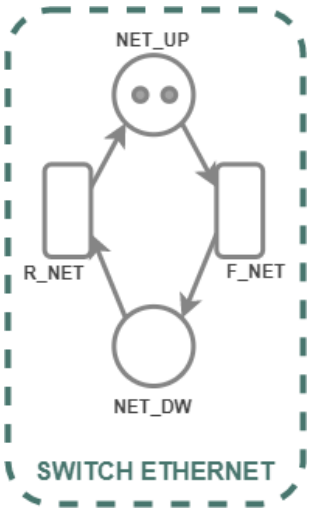


Figura 17 – Modelo em SPN representando a camada de conectividade.

A quantidade de *tokens* representa a quantidade de componentes existentes na camada e a sua localização sinaliza o estado do componente. O lugar NET_UP representa a camada no estado operacional. Por outro lado, o lugar NET_DW representa a camada no estado de falha. As transições representam as ações realizadas nos modelos. Essas ações podem ser decisões ou mudanças de estado no componente. Quando essas ações possuem um tempo fixo, são conhecidas como transições temporizadas. Caso o tempo definido nestas transições seja decorrido e as condições forem atendidas, elas são disparadas. Outra categoria de transição são as imediatas, que não possuem tempo definido, sendo disparadas quando uma determinada situação ocorre. A transição F_NET representa as falhas dos componentes e possuem os tempos médios de falha de cada componente representando os MTTFs. O mesmo raciocínio foi utilizado na transição R_NET, porém, ela representa os reparos dos componentes. Desta forma, foi adicionado o respectivo MTTRs. As transições temporizadas são configuradas com

a semântica *infinite server* para representar a operação paralela (ativo/ativo).

A seta é conhecida como arco de transição, sua responsabilidade é conectar lugares utilizando-se das transições. Ela consome e cria *tokens* dependendo de como está conectado. Caso a sua origem seja um lugar ativo, ela é responsável por consumir o *token* do lugar conectado. Caso a sua origem seja uma transição, ela gera *tokens* e coloca-os no lugar onde está conectada.

No estado inicial do modelo, o lugar NET_UP possui todos os *tokens* tornando F_NET a única transição ativa. Quando disparada, um *token* é consumido e gerado no lugar NET_DW. Agora, NET_UP e NET_DW possuem *tokens*, sinalizando que existem componentes no estado operacional e em falha. Desta forma, duas transições temporizadas estão ativas, R_NET e F_NET. Se a transição R_NET for acionada, o sistema retorna ao estado inicial com todos os *tokens* no lugar NET_UP, mas se F_NET for acionada, é gerado mais um *token* no lugar NET_DW. Caso todos os *tokens* estejam no lugar NET_DW significa que a camada está em um estado de falha.

6.1.2 Camada de virtualização

A camada de virtualização é representada pelos servidores físicos. Este trabalho adota três tipos de configurações de redundância para servidores físicos: servidores sem redundância, com redundância em *cold-standby* (ativo/passivo) e com redundância em HA (ativo/ativo). Como apresentado anteriormente, na Seção 4.1.2.2, foram escolhidas essas formas de redundâncias por serem as configurações utilizadas no ambiente de produção do sistema estudado. Independente da configuração de redundância, o equipamento servidor será sempre representado da mesma forma. Ao analisar o monitoramento de forma detalhada, foi percebido que pode-se segmentar o servidor em: componentes físicos (placa-mãe, memória, fonte de alimentação, controladores e outros componentes) e componente lógico (sistema operacional instalado no servidor).

Caso um desses componentes apresente defeito, o serviço ofertado por eles apresenta falha. Esta é a mesma característica de componentes modelados em série no RBD, desta forma, o servidor físico possui as características que permitam a sintetização.

A modelagem deste componente será hierárquica, tendo o primeiro nível em modelo RBD e o segundo nível em SPN. A modelagem de primeiro nível hierárquico está apresentada na Figura 18. Os componentes físicos estão representados pela caixa (HW) e o sistema operacional

hipervisor pela caixa (HP). Os valores de MTTF e MTTR utilizados neste modelo foram obtidos através do monitoramento e calculados através das Equações 2.9 e 2.10, respectivamente. Estes valores podem ser encontrados no Apêndice B.2.

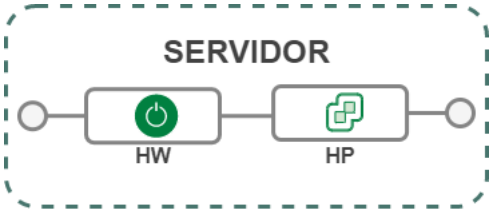


Figura 18 – Modelo em RBD representando o servidor.

Após a descoberta dos valores de MTTF e MTTR do servidor físico, através do modelo em RBD, pode-se iniciar o segundo nível da modelagem hierárquica. A partir deste momento, por ser uma modelagem hierárquica, o modelo em SPN aglutina e representa os componentes físicos e o sistema operacional do hipervisor.

Servidor sem redundância:

A estrutura física desta configuração está apresentada na Figura 19, onde pode-se perceber que podem existir vários servidores, porém, todos estão operando sem redundância entre si. Esta configuração representa servidores que não possuem componentes com um mecanismo de HA. Caso este componente esteja em um estado de falha, o sistema torna-se inacessível imediatamente.

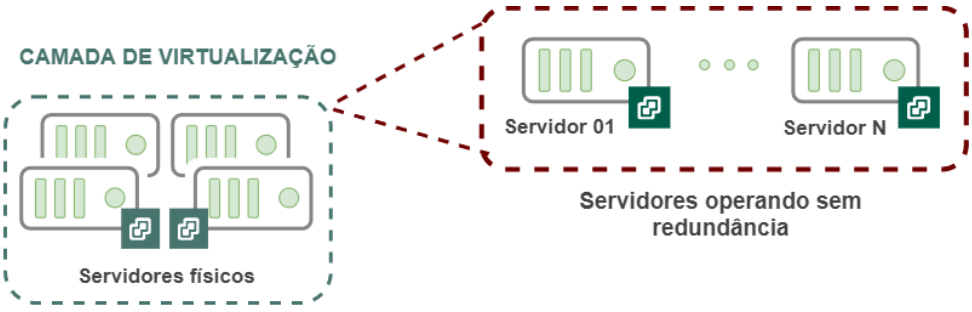


Figura 19 – Camada de virtualização - Servidores sem redundância.

Nesta configuração, as máquinas virtuais hospedadas no servidor físico não consegue realizar migrações para outros servidores. Esta dinâmica será exposta mais adiante na Seção 6.2. A representação em SPN dos servidores sem redundância está apresentada na Figura 20.

O funcionamento básico deste modelo possui semelhanças com o modelo que representa a camada de conectividade. A quantidade de *token* será sempre um, representando que existe apenas um servidor operando. O lugar SRV_UP representa o servidor no estado funcional e

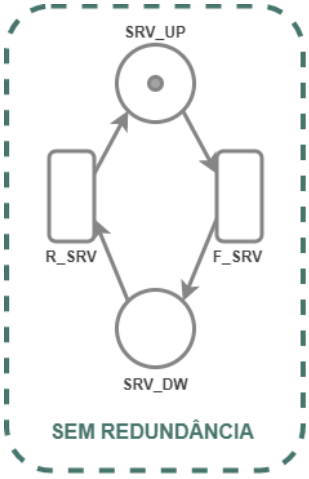


Figura 20 – Modelo em SPN representando o servidor sem redundância.

o lugar SRV_DW informa que o servidor está no estado de falha. As transições temporizadas F_SRV e R_SRV representam os tempos de médios de falha e reparo do servidor, respectivamente, e estão configuradas com a semântica *single server*. A existência de um *token* no lugar SRV_UP torna a transição F_SRV ativo, caso seja disparada, ela consome o *token* e gera um *token* no lugar SRV_DW. Neste momento, apenas a transição R_SRV está ativa, sendo disparada, o modelo volta ao estado inicial.

Servidor com redundância em HA:

A utilização de servidores com redundância na configuração de HA, ou ativo/ativo, também são utilizados neste estudo. A estrutura física desta configuração está apresentada na Figura 21. Nesta figura, pode-se perceber a existência de vários conjuntos de servidores, porém, cada conjunto possui apenas dois servidores físicos compartilhando recursos e serviços. Nesta configuração existem mecanismos de alta disponibilidade, podendo haver componentes inoperantes e mesmo assim não afetar a disponibilidade da camada.

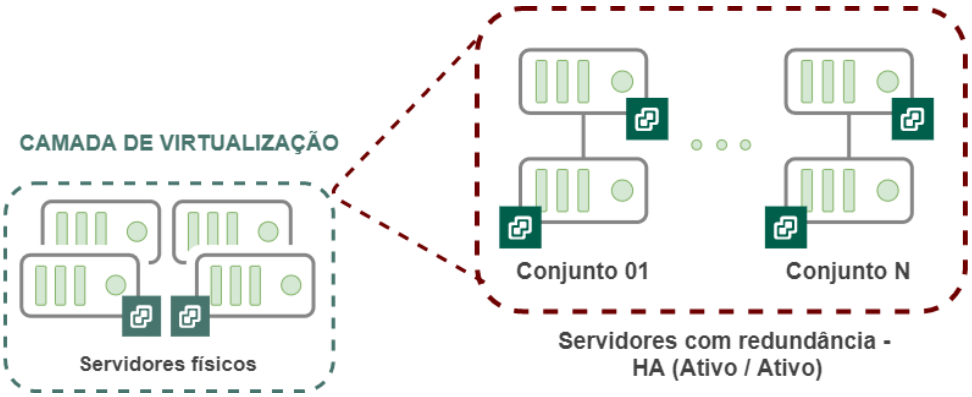


Figura 21 – Camada de virtualização - Servidores com redundância HA.

A modelagem que representa esta configuração está exposta na Figura 22. Esta configuração permite que existam migrações de VMs entre servidores do mesmo conjunto, possibilitando que os recursos entre estes servidores sejam compartilhados. O modelo representa um conjunto de servidores, tendo a quantidade de *tokens* representando a quantidade de servidores no conjunto. O funcionamento deste modelo é similar ao modelo da camada de conectividade. O serviço ofertado pelo conjunto de servidores estará disponível se existir pelo menos um *token* no lugar SRV_UP, se tornando indisponível se todos os *tokens* estiverem no lugar SRV_DW. As transições F_SRV e R_SRV são configuradas como *infinite server* e representam o MTTF e MTTR do servidor.

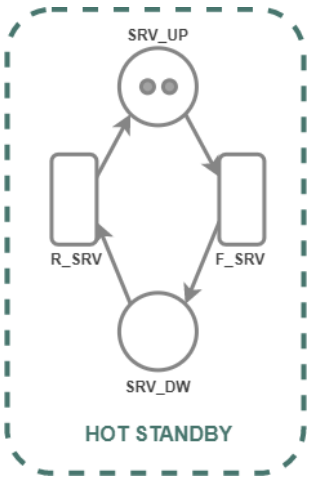


Figura 22 – Modelo em SPN representando o servidor em HA.

Servidor com redundância em *cold standby*:

A última configuração possível para representar o servidor está exposta na Figura 23, que apresenta o servidor atuando em *cold standby*. Esta configuração também possui mecanismos de redundância de componentes sendo considerado de alta disponibilidade. Entretanto, diferente do modelo ativo/ativo, os componentes utilizam o modo ativo/passivo, podendo possuir dois servidores, sendo que apenas um servidor está funcional e ativado por vez.

O modelo em SPN representando a camada de virtualização quando atuando em *cold standby* está apresentada na Figura 24. O servidor principal é identificado por SRV1 e o servidor secundário por SRV2. A modelagem apresentada na figura requer estratégias para gerenciar e identificar qual servidor está ativo e recebendo requisições externas. Assim, foi criado um lugar apresentado como E_SRV. Quando há um *token* neste local, sinaliza que o servidor principal está ativo e o servidor secundário está inativo. Também é necessário controlar a inicialização e o desligamento do servidor secundário, pois ele só pode estar em um estado ativo se o servidor

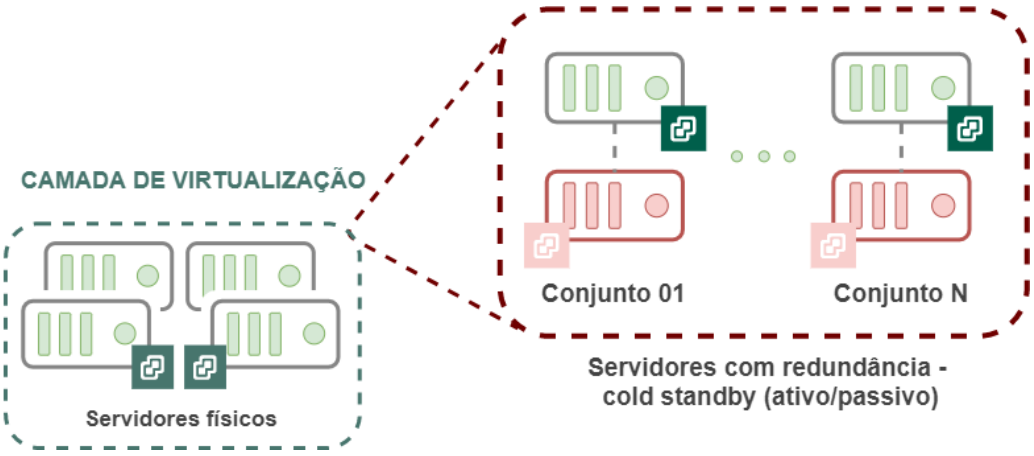


Figura 23 – Camada de virtualização - Servidores com redundância *cold-standby*.

principal estiver em um estado de falha. Estas ações de controle são realizadas pela transição temporizada START_SRV2 e pela transição imediata STDW_SRV2, respectivamente. Pode-se perceber que há uma transição imediata nesta configuração. Esta transição representa o desligamento do servidor secundário que ocorre apenas se o servidor principal estiver funcional. Esta ação não possui um tempo definido para ocorrer, sendo realizada sempre que os servidores secundários e principal estiverem no estado funcional.

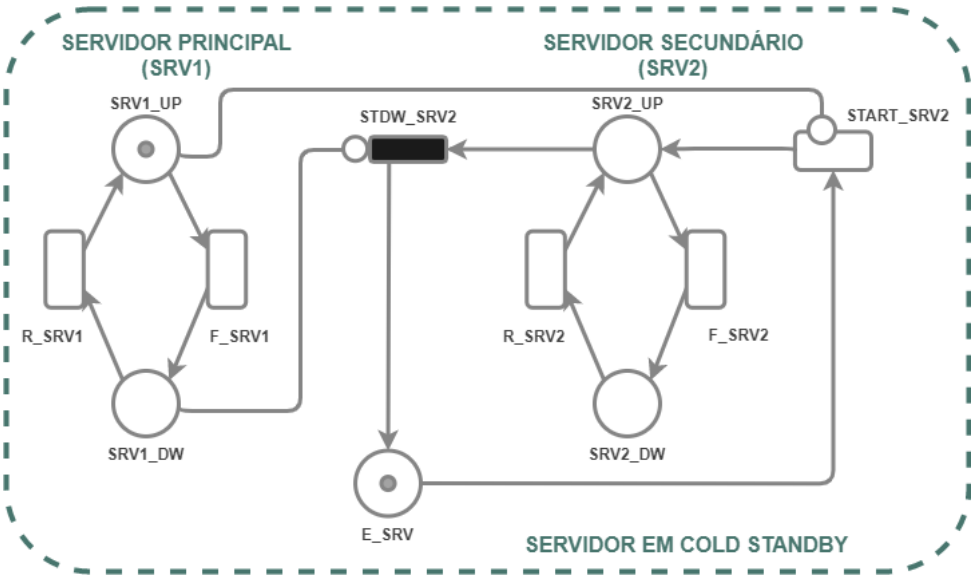


Figura 24 – Modelo em SPN representando o servidor em *cold standby*.

Por existir apenas um servidor principal e um secundário, as transições temporizadas são configuradas com a semântica *single server*. Além disso, o modelo apresentado inclui arcos de inibição e expressões de guarda para restringir os momentos de disparo das transições. A transição imediata STDW_SRV2 sofre ações do arco de inibição, do arco de transição e da expressão de guarda. O arco inibidor não permite que a transição seja disparada se houver

token no lugar SRV1_DW. Já o arco inibidor só permite que a transição seja disparada se houver um token no lugar SRV2_UP. Por último, a expressão de guarda $(\#STF\_UP = 0)\, AND\, (\#SFT\_DW = 0)$ restringe o disparo da transição, permitindo o disparo apenas se o número de *token* nos lugares STF_UP e STF_DW forem iguais a 0. Assim, a transição só será acionada se todas as restrições forem atendidas. Os lugares SFT_UP e SFT_DW representam componentes que funcionam na camada de virtualização e serão expostas mais adiante no trabalho ao apresentarmos os modelos para a estrutura lógica. Estes componentes foram apresentados na Seção 5.3 e seus respectivos modelos serão expostos na Seção 6.2. Outra transição que também sofre interferência de um arco inibidor é START_SRV2. Assim, ela não pode ser disparada enquanto houver *token* no lugar SRV1_UP.

No estado inicial do modelo *cold standby*, o servidor principal está ativo e o servidor secundário desligado, de forma que os lugares SRV1_UP e E_SRV possuem um *token* cada. A única transição ativa é F_SRV1, representando a falha do servidor principal. Quando disparada, um *token* é consumido pelo arco de transição e gerado no lugar SRV1_DW. Este momento representa uma indisponibilidade no serviço ofertado, pois todos os servidores estão inoperantes. Atualmente, as transições R_SRV1 e START_SRV2 estão ativas. Se R_SRV1 for disparada, o sistema retornará ao estado inicial com o servidor principal ativo.

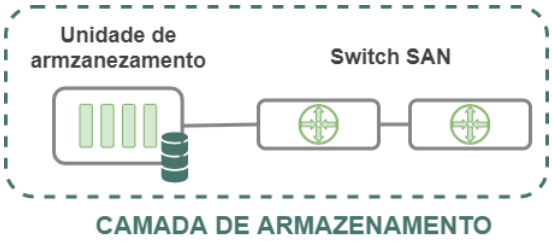
No entanto, se a transição START_SRV2 for disparada, o *token* de E_SRV é consumido e gerado no lugar SRV2_UP. Neste momento, o servidor secundário estará em um estado ativo e o servidor primário estará em um estado de falha. A partir de então, um processo externo será iniciado. A camada de orquestração migra as máquinas virtuais para o servidor secundário. As transições ativas são F_SRV2 e R_SRV1. Se F_SRV2 for disparado, o *token* será consumido e gerado em SRV2_DW, definindo que todos os servidores estão inativos e a camada de virtualização está em um estado de falha. Entretanto, se R_SRV1 for acionado, o lugar SRV1_UP receberá um *token* informando que o servidor principal voltou a atividade.

A partir deste momento duas possibilidades podem ocorrer. Uma interna e outra externa à camada de virtualização. A possibilidade interna é o disparo da transição R_SRV2 tornando o servidor secundário inoperante e consequentemente todas as VMs hospedadas neste servidor serão automaticamente desligadas. Assim, a camada de orquestração inicia um processo de *cold migration* para iniciar as máquinas virtuais para o servidor principal. A segunda possibilidade é a externa, onde o orquestrador inicia a migração das máquinas virtuais sem perda de conectividade, chamada de *live migration*. Após este processo a transição STDW_SRV2 é acionada imediatamente desligando o servidor secundário. Neste momento, o modelo volta ao

estado inicial.

6.1.3 Camada de armazenamento

Essa seção apresenta os modelos que representam a camada de armazenamento. Como apresentado anteriormente, esta camada é composta pela Unidade de Armazenamento e o cluster de Switches SAN. A figura que representa a camada de armazenamento é novamente apresentada visando melhorar o entendimento. Informações sobre o modelo desta camada podem ser encontradas no Apêndice B.1.



Como informado anteriormente, os componentes *switch SAN* e a unidade de armazenamento foram modelados diretamente em SPN. A representação do *switch SAN* utiliza da estrutura em SPN para componentes que operam em HA. Por outro lado, a camada de armazenamento utiliza a estrutura para apenas um componente. Os modelos podem ser conferidos na Figura 25, tendo a representação do *switch SAN* na Figura 25a e camada de armazenamento na Figura 25b.

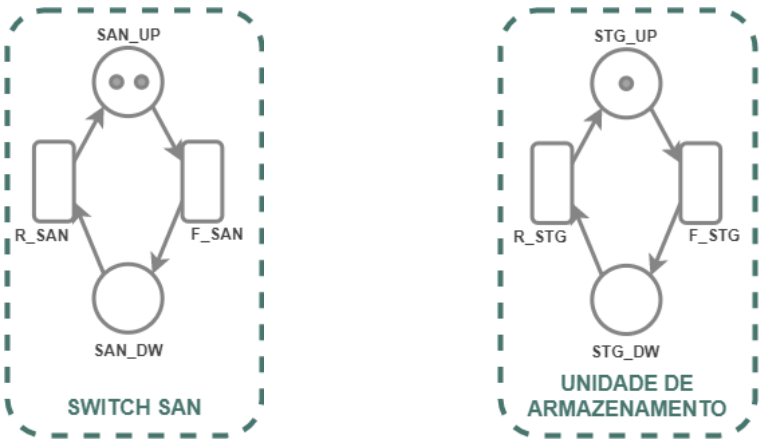


Figura 25 – Modelo em SPN representando as camadas de conectividade e armazenamento.

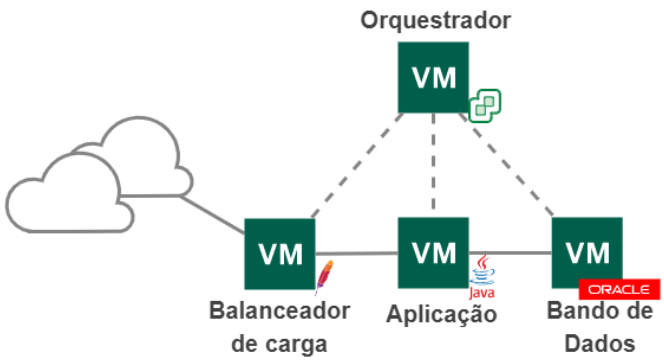
O *modus operandi* dos modelos é o mesmo dos já apresentados. O número de *tokens* indica a quantidade de componentes. Os lugares SAN_UP e STG_UP representam os componentes

no estado operacional e os lugares SAN_DW e STG_DW representam os componentes no estado de falha. O serviço ofertado pelo *switch SAN* está inoperante quando todos os *tokens* estiverem no lugar SAN_DW. As transições representam as ações realizadas nos modelos, sendo as transições F_SAN e F_STG a representação das falhas dos componentes, possuindo os tempos médios de falha de cada componente. Já as transições R_SAN e R_STG representam os reparos dos componentes, sendo adicionados seus respectivos MTTRs. No modelo da unidade de armazenamento, as transições são configuradas com a semântica *single server*, e no modelo do *switch SAN* como *infinite server* para representar a operação paralela.

Este modelo apresenta uma peculiaridade por conter dois tipos de componentes na camada, *switch SAN* e unidade de armazenamento. A camada só estará no estado operacional se os dois modelos possuírem *tokens* nos lugares que representam os componentes operacionais (SAN_UP e STG_UP). Caso um dos modelos não apresentem tokens nestes lugares, toda a camada de armazenamento estará indisponível.

6.2 ESTRUTURA LÓGICA

Nesta seção são apresentados os modelos hierárquicos utilizados na representação da estrutura lógica do sistema estudado. Para facilitar o entendimento, a Figura 15 é reapresentada para mostrar a estrutura lógica exposta na Seção 5.3. A estrutura lógica é o ambiente virtual utilizado na solução. Todos os componentes desta camada são VMs, chamada neste estudo por *camada de software*. A camada de *software* foi agrupada segundo as funcionalidades dos seus componentes. Assim, foram criadas quatro subcamadas: **Orquestração (VC), Balanceador de Carga (LB), Aplicação (APP) e Banco de Dados (BD)**.



Cada subcamada possui dois componentes: o SO e a aplicação responsável por fornecer um serviço. Através do monitoramento foi identificado que o SO sofre interferência direta do

serviço que está hospedando. Desta forma, os parâmetros de MTTF e MTTR do SO que hospeda o balanceador de carga são distintos dos parâmetros de SO que hospeda o banco de dados, por exemplo. Pensando nesta característica, foram utilizados valores individuais para cada SO usado neste estudo. Estes valores serão expostos mais adiante.

Seguindo esta linha de observação, a configuração desta estrutura se assemelha com a camada de virtualização, permitindo a modelagem hierárquica. Esta diretriz é possível pelo fato da relação entre os dois componentes de cada subcamada ser de dependência, pois o serviço ofertado pela subcamada não funciona se um dos componentes estiver no estado de falha. Assim, foram construídos modelos distintos em RBD para cada subcamada. Em seguida foram utilizados os resultados do modelo como insumo para o modelo em SPN, criando, assim, outra estrutura de modelos hierárquicos de dois níveis. A identificação da quantidade dos componentes e suas respectivas redundâncias também foram encontradas na Macro-atividade 4.1, exposta na Metodologia 4.1 e apresentadas na Tabela 6.

Tabela 6 – Relação dos componentes da estrutura lógica.

Serviço	**Componente(s)**	**Tipo de redundância**
Orquestração (VC)	Orquestrador (VCenter)	✗
Balanceador de carga (LB)	SO do Balanceador (SO_{ba}) Serviço do Balanceador (SER_{ba})	Série
Aplicação (APP)	SO da Aplicação (SO_{ap}) Serviço da Aplicação (SEO_{ap})	Série
Banco de dados (BD)	SO do Banco de dados (SO_{db}) Serviço do Banco de dados (SER_{db})	Série

A modelagem utilizada para esta estrutura será apresentada de uma forma diferente da exposta para a camada computacional. Para a estrutura lógica, as modelagens hierárquicas do primeiro nível são apresentadas na seção 6.2.1 e as modelagens de segundo nível na Seção 6.2.2. Esta apresentação foi utilizada por semelhança muito grande entre as modelagens das subcamadas de *software*.

6.2.1 Primeiro nível hierárquico

O primeiro nível hierárquico, modelagem em RBD, está exposto na Figura 26. Nele, pode-se perceber: o modelo da camada de orquestração apresentado na Figura 26a, o modelo da

camada de balanceador de carga na Figura 26b, o modelo da camada de aplicação na Figura 26c e o modelo da camada de banco de dados no Figura 26d. Em todos os modelos, existem caixas com a nomenclatura SO, que representam os Sistemas Operacionais dos servidores e outra caixa contendo a nomenclatura do serviço.

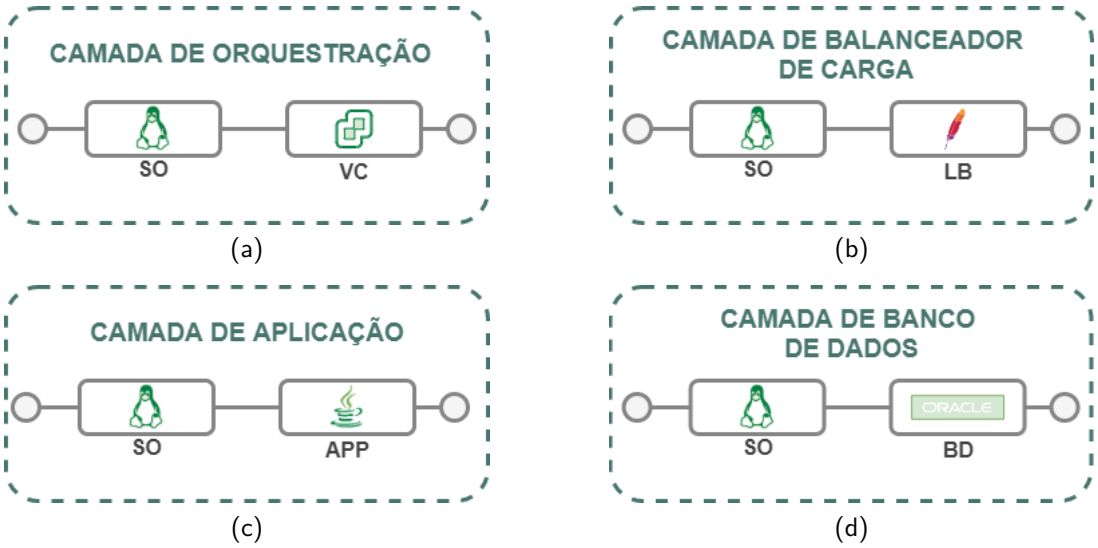


Figura 26 – Modelos em RBD representando a camada de *software*.

Os valores de MTTF e MTTR obtidos nos modelos que representam as subcamadas, serão utilizados como insumos na modelagem de segundo nível hierárquico, em SPN. Informações sobre os modelos das camadas virtuais podem ser encontradas no Apêndice B.

6.2.2 Segundo nível hierárquico

Essa seção mostra os modelos em SPN que representam os componentes virtuais da camada de *software*. Todos os modelos aqui apresentados são do segundo nível da modelagem hierárquica, que recebem como insumo a saída dos modelos em RBD. As estruturas de modelagem em SPN são as mesmas apresentadas e explicadas anteriormente na Seção 2.5.2.

Por serem componentes virtuais, as subcamadas aqui apresentadas sofrem interferências externas das camadas de virtualização e armazenamento. Se alguma dessas camadas estiver em estado de falha, a camada de *software* será diretamente afetada e forçada a também entrar no estado de falha. Por isso, pode ser classificada como sensível a mudanças externas de estados.

Os servidores são a base de funcionamento da camada lógica. Assim, entende-se que as máquinas virtuais estão hospedadas na camada de virtualização. Com isso, sempre que

houver um servidor no estado funcional, haverá máquinas virtuais hospedadas nele. A sua inoperabilidade resulta no desligamento automático das máquinas virtuais. Caso a camada de virtualização possua redundância, o orquestrador irá identificar a falha no servidor e iniciar um processo de migração das máquinas virtuais para um servidor no estado operacional. Esta migração é realizada após a falha da máquina virtual, sendo chamada de *cold migration*. Outra possibilidade de migração é a *live migration*, que pode ocorrer em configurações *cold-standby*. A camada de orquestração identifica que o servidor principal se recuperou após uma falha, resultando em dois servidores ativos. Então, o orquestrador realiza a migração das máquinas virtuais do servidor secundário para o principal sem haver interrupção nos serviços ofertados pela máquina virtual.

Este estudo apresenta três configurações distintas para camada de *software*: sem redundância, HA e *cold standby*.

Camada de *software* sem redundância e HA:

A Figura 27 expõe os modelos representando uma camada de *software* sem redundância e em HA. Semelhante aos modelos de estrutura computacional, a configuração sem redundância, exposta na Figura 27a, representa as camadas que não possuem alta disponibilidade nos seus componentes. Por outro lado, a configuração HA, apresentada na Figura 27b, representa as camadas com mecanismos de alta disponibilidade no modo ativo/ativo.

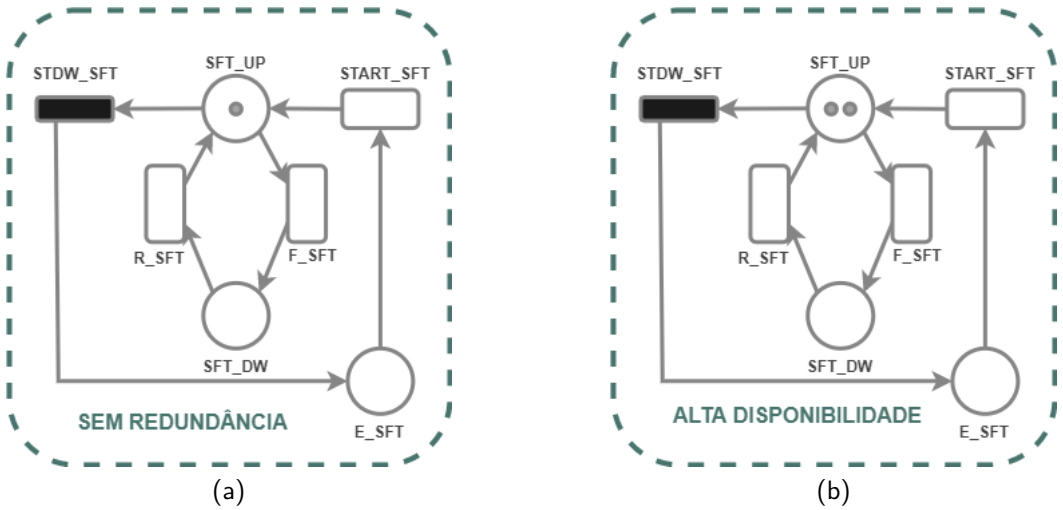


Figura 27 – Modelo em SPN da camada de *software* sem redundância e em HA.

O rótulo (SFT) foi utilizado nos modelos para representar uma camada de *software*. Uma representação genérica pelo fato do modelo poder representar qualquer subcamada de *software*. No lugar onde aparece esta nomenclatura, pode-se ler VC, LB, APP ou DB representando

rótulos usados nos modelos para identificar as camadas de orquestração, balanceador de carga, aplicação e banco de dados, respectivamente.

O modelo apresentado na Figura 27a foi utilizado nas camadas de orquestração e banco de dados, pois utilizam apenas um componente. Por outro lado, o modelo exposto na Figura 27b foi utilizado para representar as camadas de balanceador de carga e aplicação, por possuírem mais de um componente operando em HA. A quantidade de *tokens* utilizada nos modelos representam a quantidade de componentes existentes em cada subcamada.

Os lugares SFT_UP e SFT_DW representam a camada no estado operacional e de falha, respectivamente. As transições temporizadas F_SFT e R_SFT representam os MTTFs e MTTRs das camadas. Para modelos sem redundância, essas transições são configuradas com a semântica *single server* e para modelos em HA, utiliza-se *infinite server*.

Como mencionado um pouco acima, as camadas modeladas nesta seção são sensíveis a mudanças externas. Quando ocorrem essas interferências, todo o funcionamento natural do sistema é alterado. A camada não pode tornar-se funcional caso outros componentes do ambiente estejam em falha. Desta forma, é preciso identificar corretamente o momento destas intervenções para podermos modelar os cenários possíveis. Pensando nisto, foi criado um lugar, E_SFT, responsável por identificar estas intervenções. Assim, sempre que houver *token* no lugar E_SFT representa que a camada sofreu alguma interferência externa.

O controle da entrada e saída da camada neste estado é realizado por duas transições STDW_ SFT e START_SFT. A transição responsável por colocar a camada no estado de interferência externa é o STDW_SFT, realizando o desligamento da camada sempre que ocorre as intervenções externas. Esta transição é do tipo imediata, pois não possui tempo definido de disparo. Ela é acionada através de gatilhos ou expressões de guarda. Estas expressões estão expostas na Tabela 7.

Tabela 7 – Expressões de guarda da camada de *software*.

Transição	**Expressão de guarda**
R_SFT	$(\#SAN\_UP > 0)$ and $(\#STG\_UP > 0)$ and $(\#SRV\_UP > 0)$
STDW_SFT	$(\#SAN\_UP = 0)$ or $(\#STG\_UP = 0)$ or $(\#SRV\_UP = 0)$
START_SFT	$(\#SAN\_UP > 0)$ and $(\#STG\_UP > 0)$ and $(\#SRV\_UP > 0)$
R_APP / START_APP	$(\#SAN\_UP > 0)$ and $(\#STG\_UP > 0)$ and $(\#SRV\_UP > 0)$ and $(\#BD\_UP > 0)$

Por outro lado, a transição responsável por retirar a camada deste estado é a START_SFT.

Ela também só pode ser disparada em condições restritas e possui expressões de guardas como gatilhos, também expostas na Tabela 7. Porém, os serviços modelados não podem se tornar funcional imediatamente por precisar de tempo para que todo o sistema seja iniciado, sendo assim, uma transição temporizada.

Outra transição que possui expressão de guarda é R_SFT. Este procedimento é necessário para que a camada só volte ao estado operacional se os componentes físicos estiverem funcionais. A informação sobre a guarda também está na Tabela 7.

Analisando as expressões de guarda, pode-se perceber que existem componentes externos à camada de *software*. A expressão SAN remete aos *Switchs SAN*, a expressão STG é referente à unidade de armazenamento e por último, SRV se refere ao servidor físico onde a camada de *software* está hospedada. Apenas uma expressão apresentada na tabela de guardas é interna à camada de *software*, o BD. Este termo se refere a camada de banco de dados e só aparece na modelagem da camada de aplicação. Existe uma correlação direta da aplicação com o banco de dados. Na inicialização da aplicação, informações são buscadas no BD. Sem essas informações a aplicação não se torna operacional. Desta forma, ela só pode iniciar caso a BD esteja operacional.

O funcionamento dos dois modelos, sem redundância e HA, são semelhantes aos modelos de estrutura computacional. No estado inicial, o único lugar que possui *tokens* é SFT_UP e a única transição ativa é F_SFT. Quando disparada, um *token* é consumido de SFT_UP e gerado em SFT_DW. No modelo sem redundância, a camada está em estado de falha e a única transição ativa é a R_SFT, caso a expressão de guarda seja atendida. Se a expressão não for atendida nenhuma transição fica ativa. Se a transição R_SFT for disparada, o *token* é consumido e gerado em SFT_UP restaurando o modelo ao estado inicial. Já no modelo HA, neste momento, os lugares SFT_UP e SFT_DW possuem *tokens* sinalizando que existem componentes nos estados de falha e operacional. Desta forma, a transição F_SFT está ativa e a transição R_SFT pode estar ativa caso a expressão de guarda esteja sendo atendida, caso contrário estará desativada. No caso da transição R_SFT estar ativa e for disparada, o sistema retorna ao estado inicial com todos os *tokens* em SFT_UP. Em contrapartida, caso a transição F_SFT seja disparada, mais um *token* é gerado em SFT_DW. Nas camadas que possuem dois componentes, ficarão no estado de falha. Porém, em camadas com mais componentes o funcionamento continua o mesmo e a camada estará no estado de falha caso todos os *tokens* estejam no lugar SFT_DW.

A qualquer momento, se houver *token* no lugar SFT_UP e a expressão de guarda para a

transição imediata STDW_SFT for atendida, ela será disparada. Assim, um *token* é consumido e gerado em E_SFT, sinalizando que a camada está no estado de falha provocada por interferência externa. A partir de então, a única transição que pode disparar é START_SFT se a interferência externa tiver sido corrigida, ou seja, se a expressão de guarda for atendida. Caso isto ocorra e a transição START_SFT for disparada, um *token* é gerado em SFT_UP e o sistema retorna a um estado funcional.

Camada de *software* com redundância em *Cold standby*:

A última configuração que representa os modelos da camada de *software* é a *cold standby*, apresentada na Figura 28. Esta configuração precisa ser hospedada em um servidor que também esteja operando em *cold standby*. O modelo foi projetado para permitir a identificação de qual servidor físico a máquina virtual está hospedada. Este cenário sofre a mesma interferência externa dos modelos anteriores, então, também foi criado o lugar chamado E_SFT para controlar estes casos.

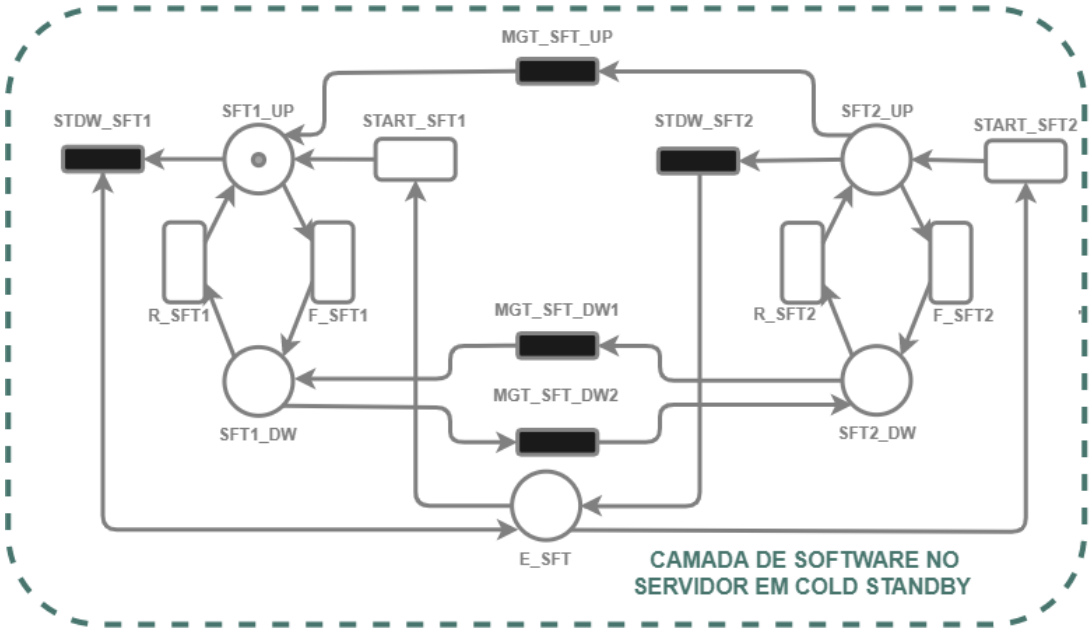


Figura 28 – Modelo em SPN representando camada virtual hospedada em servidor em *cold standby*.

As transições de entrada e saída deste estado seguem os mesmos parâmetros também apresentados anteriormente. O diferencial é que existem duas transições para cada ação. As transições que possuem no seu nome SFT1 indicam que as ações são tomadas na camada de *software* quando está hospedada no servidor principal. Já as transições que possuem no seu nome SFT2 representam as ações realizadas quando a VM está hospedada no servidor

secundário. Neste modelo também é contemplado as migrações das máquinas virtuais entre servidores.

Os lugares SFT1_UP e SFT1_DW representam o serviço hospedado no servidor principal enquanto SFT2_UP e SFT2_DW representam o serviço hospedado no servidor secundário. Os recursos básicos são semelhantes aos outros modelos já expostos. Os lugares SFT1_UP e SFT2_UP representam camadas no estado operacional e SFT1_DW e SFT2_DW no estado de falha. As transições temporizadas F_SFT1 e F_SFT2 representam os MTTFs, já as transições R_SFT1 e R_SFT2 representam os MTTRs da camada. É possível realizar duas configurações distintas para estas transições temporizadas. Caso a camada opere em um modo sem redundância, as transições são configuradas com a semântica *single server*. Em contrapartida, caso a camada opere em modo de HA, as transições são configuradas com a semântica *infinite server* e o número de *tokens* representa o número de componentes.

As transições temporizadas responsáveis por inciar e desligar a camada são: START_SFT1, START_SFT2, STDW_SFT1, STDW_SFT2, R_SFT1 e R_SFT1. Já as transições responsáveis por realizar as migrações são: MGT_SFT_UP, MGT_SFT_DW1 e MGT_SFT_DW2. Todas as transições mencionadas possuem expressões de guarda. As informações sobre as expressões de guarda são apresentadas na Tabela 8, que possuem referências ao servidor físico e à camada de armazenamento por haver correlação direta desses recursos.

Tabela 8 – Expressões de guarda da camada de *software* em *cold standby*.

Transição	**Expressão de guarda**
STDW_SFT1	$(\#SAN\_UP = 0)$ or $(\#STG\_UP = 0)$ or $(\#SRV1\_UP = 0)$
STDW_SFT2	$(\#SAN\_UP = 0)$ or $(\#STG\_UP = 0)$ or $(\#SRV2\_UP = 0)$
START_SFT1 / R_SFT1	$(\#SAN\_UP > 0)$ and $(\#STG\_UP > 0)$ and $(\#SRV1\_UP > 0)$
START_SFT2 / R_SFT2	$(\#SAN\_UP > 0)$ and $(\#STG\_UP > 0)$ and $(\#SRV2\_UP > 0)$
MGT_SFT_UP	$(\#SRV1\_UP > 0)$
MGT_SFT_DW1	$(\#SRV1\_UP > 0)$
MGT_SFT_DW2	$(\#SRV1\_UP = 0)$ and $(\#SRV2\_UP > 0)$

O modelo também representa migrações de VMs. Estas migrações podem ocorrer reativamente, após o servidor apresentar problemas ou uma *live migration*, que permite a migração da VM sem indisponibilidade do serviço ofertado. Todas essas migrações são de responsabilidade

da camada de orquestração. A transição MGT_SFT_UP realiza a *live migration*. Ela ocorre sempre que a camada de *software* está operacional no servidor secundário e o servidor principal retorna ao estado operacional. Como existe a preferência de atuar sempre no servidor principal, o orquestrador realiza a migração da VM sem que o serviço sofra interrupções. Por outro lado, as transições MGT_SFT_DW1 e MGT_SFT_DW2 representam as migrações reativas que ocorrem após a interrupção do serviço. Estas migrações possuem a responsabilidade de garantir que uma VM não tente iniciar em um servidor no estado de falha. Sempre que uma VM estiver no estado de falha e o servidor em que ela estiver hospedada também entrar no mesmo estado, será realizada a migração para outro servidor que estiver operacional. Desta forma, a VM sempre iniciará em um servidor operacional.

Igualmente aos modelos sem redundância e em HA, este modelo possui dois caminhos possíveis. O caminho interno e o de interferência externa. Falhas nas camadas de virtualização ou armazenamento podem acionar o caminho de interferência externa a qualquer momento.

Assumindo que o estado inicial é representado pelo servidor primário no estado operacional e a camada virtual hospedada nele. Desta forma, o lugar SFT1_UP possui um *token*, e a única transição ativa é F_SFT1. Se acionada, um *token* é gerado em SFT1_DW. Neste ponto, a única transição possível é R_SFT1 que, ao ser acionada, é gerado um *token* em SFT1_UP, retornando ao estado inicial do modelo.

Neste momento são apresentados os possíveis caminhos para interferência externa. Todas as ações aqui explicadas consideram que ocorreram problemas nas camadas de armazenamento ou virtualização. Caso haja um *token* no lugar SFT1_UP, a transição imediata STDW_SFT1 pode ser disparada e gerar um *token* no lugar E_SFT. Agora, a camada de *software* está em um estado de falha devido a ações externas. Quando as interferências externas são corrigidas, as transições START_SFT1 ou START_SFT2 são disparadas. O START_SFT1 é acionado se o servidor principal estiver operacional, retornando o modelo ao estado inicial. Já a transição START_SFT2 é disparada caso o servidor secundário esteja funcional. A transição gera um *token* em SFT2_UP, sinalizando que no momento o servidor secundário está hospedando a camada de *software*. Neste ponto, as transições F_SFT2 e R_SFT2 podem executar o caminho interno representando falha e restauração dos componentes, respectivamente. A transição imediata STDW_SFT2 também pode realizar o caminho de interferência externa, caso o servidor secundário ou a camada de armazenamento falhe, retornando o *token* para o lugar especial E_SFT.

6.3 ESTRUTURA DO CENTRO DE DADOS EM SPN

Essa seção descreve os modelos de disponibilidade que representam as estruturas DCS e Estrutura do Sistema (SS). A DCS representa os componentes responsáveis pela refrigeração e distribuição de energia do CD. O seu funcionamento foi descrito detalhadamente na Seção 5.1. Já a SS é uma síntese das estruturas computacionais e lógicas, descritas nas Seções 5.2 e 5.3, respectivamente. A estratégia de utilizar uma sintetização nos modelos em SPN foi adotada para possibilitar uma abstração maior dos componentes, já que neste momento estamos tratando de estruturas como o CD. Desta forma, consegue-se abstrair todas as camadas de virtualização e *software* para a utilização de apenas um modelo que as representem. Ainda nesta seção, outra simplificação em SPN para representação do CD em uma estrutura única é adotada. Neste caso, o CD hospeda muitos sistemas, mas essa informação foi abstraída visando a construção de um modelo de disponibilidade que represente o sistema estudado com a estrutura física do CD.

O objetivo principal desta seção é descrever a criação de um modelo de disponibilidade que represente o sistema, incluindo a estrutura do CD e contemplando a ocorrência de problemas externos ao CD. Estes problemas externos ao CD são situações que fogem do controle dos administradores do sistema e do CD, mas que impactam diretamente as suas disponibilidades. Para estes problemas externos, a nomenclatura CCF é utilizada. Estas CCFs podem variar desde uma simples interrupção no fornecimento elétrico, a um desastre natural. Algumas CCFs estão expostas na Seção 2.1. A estratégia de uma modelagem simplificada possibilita uma visão de todo o ambiente, além de servir como base para o próximo modelo que será a replicação do CD, exposta na Seção 6.4.

A estratégia da simplificação consiste em sintetizar estruturas complexas e com um alto custo computacional no seu processamento em um modelo simples e de baixo consumo computacional. Nos modelos em SPN apresentados neste estudo, os valores de MTTF e MTTR foram usados para calcular o valor da disponibilidade. Estes valores são encontrados através do monitoramento do ambiente ou de modelos em RBD. Na simplificação segue-se o mesmo preceito, sendo necessário informar os valores de MTTF e MTTR. O valor do MTTR de estruturas em série pode ser encontrado com o somatório de todos os MTTRs. A variável faltante é o valor de MTTF. Para encontrá-lo a Equação 2.12 é adotada, onde o valor da disponibilidade é a métrica encontrada no modelo que deseja-se simplificar. Assim, são conseguidos os valores de MTTF e MTTR para aplicarmos no modelo em SPN.

A seguir, está sendo apresentado o modelo refente ao DCS e a primeira sintetização (SS). Nesta sintetização, aglutinaram-se as camadas de virtualização e *software* em um único componente básico em SPN. A Figura 29 representa o modelo de disponibilidade criado. O grupo classificado como DCS representa a estrutura física do DC, já a grupo da Estrutura do Sistema está representada por SS.

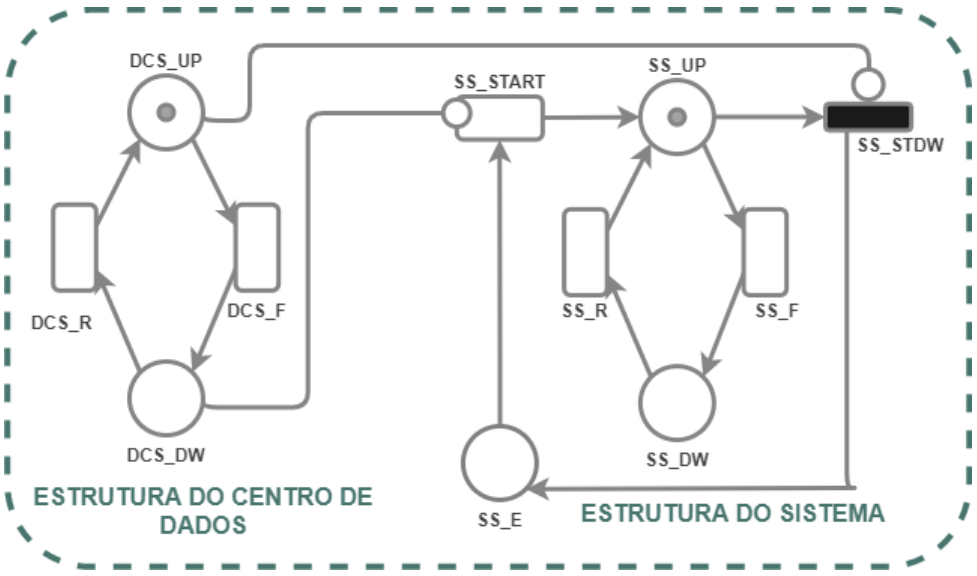


Figura 29 – Modelo em SPN representando estruturas do Centro de dados e do Sistema.

A estrutura SS sofre interferência direta do DCS. Visando a identificação do momento em que a SS entra no estado de falha causado por falha de DCS, foi utilizada a mesma estratégia já apresentada sendo criado o lugar chamado SS_E. Com esta estratégia é possível controlar a inicialização e desligamento da SS, pois ela só pode estar em estado funcional se o DCS estiver em estado operacional. Essas ações são realizadas pela transição temporizada SS_START e pela transição imediata SS_STDW, respectivamente. As transições temporizadas têm a semântica *single server*. Além disso, o modelo apresentado inclui arco de inibição que restringe o disparo destas transições. A inibição não permite o disparo das transições, enquanto houver *token* no lugar onde está conectado. Por outro lado, o arco de transição só permite o disparo quando houver *token* no lugar conectado. Portanto, a transição só é disparada quando todas as restrições forem atendidas.

No estado inicial, os lugares DCS_UP e SS_UP possuem um *token* cada. As transições ativas são DCS_F e SS_F. Se SS_F for acionada, um *token* será consumido e gerado em SS_DW sinalizando que o sistema está no estado de falha. Neste momento, a SS_R e DCS_F estão ativos. Se a SS_R for acionado, o sistema retorna ao estado inicial. No entanto, se DCS_F for disparado, DCS_DW receberá um *token*. Atualmente, o DCS e o SS estão em um

estado de falha. A partir de agora, a única transição ativa é DCS_R. Mesmo com um *token* em SS_DW, a transição SS_R não pode ser disparada por possuir a expressão de guarda $(\#DCS\_UP > 0)$, que só permite o acionamento se DCS estiver no estado operacional. Quando DCS_R é disparado, o DCS volta a um estado funcional e a transição SS_R pode ser disparada voltando ao estado inicial do modelo. Mais informações sobre o referido modelo estão no Apêndice B.7.

O segundo modelo usado para representar o CD é mostrado na Figura 30. Simplificou-se o modelo anterior (Figura 29) sendo adicionado a capacidade de interagir com falhas externas ao CD (CCF). O CCF pode ser uma interrupção do acesso externo, falta de energia, desastres naturais, entre outros problemas que interrompem o acesso externo ao sistema. Foi utilizado o termo Causas Comuns de Reparo (CCR) para representar a ação de retorno do CD ao estado funcional após um CCF. Neste modelo, foram consolidados os grupos DCS e SS em um único sistema básico em SPN. Desta forma, ao identificar que o CD está em estado operacional significa que o DCS e SS estão operacionais. Este modelo representa o CD e a nomenclatura Centro de Dados único é utilizada pelo fato de apresentar outras estruturas adiante no trabalho sobre replicação do CD.

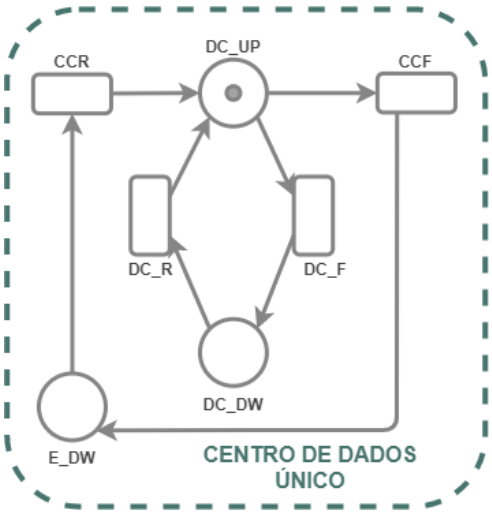


Figura 30 – Modelo em SPN representando o Centro de Dados único.

Os lugares E_DW e DC_DW representam o CD no estado de falha. O lugar DC_DW significa que o CD falhou devido a problemas internos, enquanto E_DW é devido a falhas externas. As transições são configuradas com a semântica *single server*. Um *token* é gerado em E_DW sempre que ocorre um CCF, indicando que o CD não está disponível devido a um problema externo. No estado inicial, o único *token* está em DC_UP, sinalizando que o CD está funcional. As transições ativas são CCF e DC_F, não podem ser disparadas simultaneamente

e no momento dos seus disparos, um *token* é consumido de DC_UP e gerado em E_DW ou DC_DW. A partir de agora, as transições ativas são CCR ou DC_R. Quando disparadas, o DC_UP recebe um *token*, retornando ao estado inicial do modelo.

6.4 REPLICAÇÃO DO CENTRO DE DADOS EM SPN

A última estrutura em SPN apresentada neste capítulo representa a replicação do CD. Como visto na Seção 2.1, na replicação de CD pode-se utilizar configurações como: ativo/ativo ou ativo/passivo. Cada estrutura de replicação possui características e necessidades distintas. Porém, para ambas configurações, necessita-se de uma unidade de controle que identifique qual o CD está ativo e recebendo os acessos externos. O modelo utilizado está exposto na Figura 31. Nele, pode-se perceber a unidade de controle, destacada no grupo denominado Controle do Centro de Dados, e as referências dos CDs ativo e passivo, nos grupos Centro de dados principal e secundário.

Este estudo foi baseado em dados reais extraídos do ambiente estudado. Porém, não existe uma replicação de CD para realizar o monitoramento. Desta forma, foram replicadas as configurações do CD.Os parâmetros encontrados até o momento são utilizados como referência sendo aplicados no CD secundário. Assim, no estudo existem dois CD idênticos. O Centro de Dados Principal (PDC) corresponde a Figura 31b, conjunto na caixa tracejada vermelha. Já o Centro de Dados Secundário (SDC), Figura com ilustração tracejada de amarelo, está representado na Figura 31c. A unidade de controle está representada no conjunto na caixa tracejada verde, Figura 31a. Quando um CD estiver classificado como Ativo, pode-se entender que ele está recebendo o tráfego de acesso externo ao ambiente tendo seus recursos físicos utilizados. Outra característica é que os CDs atuam independentemente. Eles podem estar operacionalizando de forma simultânea, porém, apenas o CD ativo recebe solicitações de acesso.

Quando há um *token* em P_A, sinaliza que o PDC está ativo. Se o *token* estiver em S_A, significa que o SDC está ativo. O lugar classificado como DC_DW representa que o CD que estava ativo, apresentou falha e, neste momento, nenhum CD está recebendo as requisições de acesso. As transições imediatas P_START e S_START são responsáveis por redirecionar o tráfego de acesso para o CD ativo; já as transições P_STDW e S_STDW param o tráfego de acesso do CD ativo. Para garantir o funcionamento adequado nestas transições, foram configuradas expressões de guarda conforme a Tabela 9. Mais informações sobre o referido

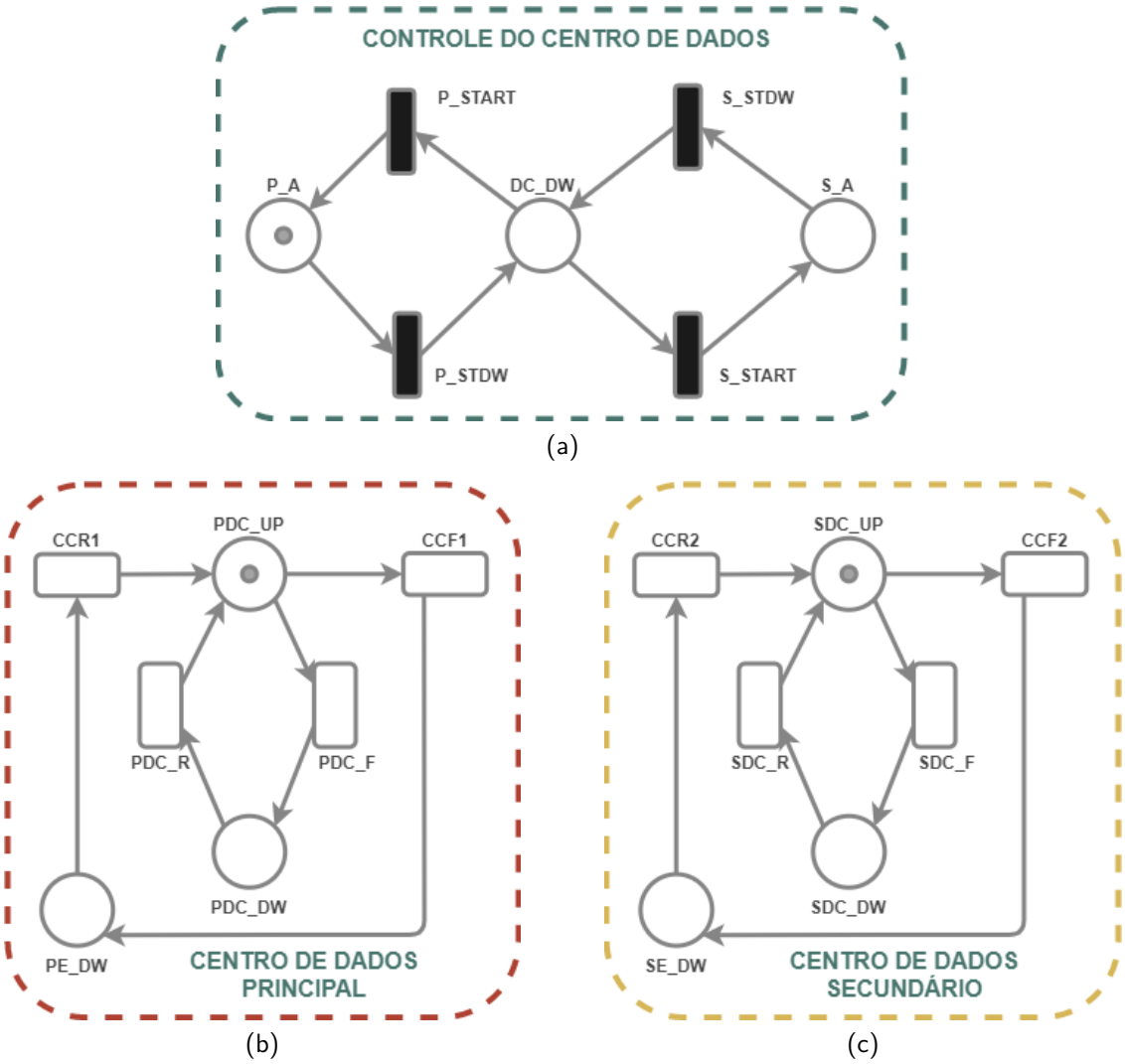


Figura 31 – Modelo em SPN representando as camadas de conectividade e armazenamento.

modelo estão no Apêndice B.8.

Tabela 9 – Expressões de guarda do controle de Centro de dados.

Transição	**Expressão de guarda**
P_STDW	$(\#PDC\_UP < \#P\_A)$
S_STDW	$(\#SDC\_UP = 0)$
P_START	$(\#PDC\_UP > \#P\_A)$
S_START	$((\#SDC\_UP = 1)$ and $(\#S\_A = 0))$

A expressão de guarda adicionadas na transição imediata P_STDW permite o seu disparo apenas quando a quantidade de *tokens* em PDC_UP for menor que o número de *tokens* em P_A. Esta expressão de guarda não permite que um PDC, ao estar no estado de ativo, deixe de ser ativo sem que esteja no estado de falha. A expressão de guarda do lugar S_STDW

assegura a retirada do estado de ativo de um SDC quando o mesmo apresenta falha. No lugar P_START, existe uma expressão de guarda que permite tornar ativo um PDC quando o mesmo estiver no estado funcional. A última expressão de guarda do controle do centro de dados está no lugar S_START, garantindo que apenas um SDC pode se tornar ativo.

No estado inicial, existem *tokens* em P_A, PDC_UP e SDC_UP. As transições de falha ativas são CCF1, PDC_F, CCF2 e SDC_F. Quando disparada alguma destas transições, um *token* é adicionado ao lugar de estado das falhas PE_DW, PDC_DW, SE_DW ou SDC_DW, receptivamente. Se o CCF1 for disparado, sinalizar que um CCF ocorreu no PDC e o tornou indisponível. Neste momento, a transição imediata P_STDW é disparada gerando um *token* em DC_DW. Depois disso, caso as expressões de guardas sejam atendidas, a transição S_START é disparada tornando o SDC ativo e recebendo o tráfego de acesso. No momento, existem *tokens* nos locais PE_DW, SDC_UP e S_A, e as transições ativas são CCR1, SDC_F e CCF2. Caso SDC_F ou CCF2 sejam acionados, é gerado um *token* no SDC_DW ou SE_DW. Independentemente da transição disparada, o SDC está em um estado de falha e o sistema está completamente inacessível. No entanto, se o CCR1 for disparado, será gerado um *token* em PDC_UP, indicando que os dois CDs estão operacionais, mas o SDC ainda está ativo. Desta forma, se houver problemas no SDC, é acionada a transição S_STDW e depois a P_START, tornando o PDC ativo e acessível externamente.

7 ESTUDOS DE CASOS

Neste capítulo são apresentados os Estudos de Casos realizados neste trabalho. Cada Estudo de Caso possui objetivos específicos, porém, todos convergem para o propósito de aprimorar a disponibilidade do sistema analisado. Este capítulo apresenta cinco Estudos de Casos: disponibilidade da *baseline*; avaliação de disponibilidade do sistema; evolução da disponibilidade; avaliação de disponibilidade do centro de dados; e, disponibilidade do centro de dados redundantes. Estes estudos seguem a estrutura apresentada na Metodologia 4.

A estrutura deste capítulo segue o caminho percorrido nos Estudos de Casos deste trabalho. Os primeiros são sobre análises e sugestões da estrutura física e lógica. Neles, são identificados pontos de possíveis melhorias na infraestrutura interna do CD, camadas virtuais e servidores físicos. Em seguida, é realizada uma análise da melhoria da disponibilidade alcançada com os cenários sugeridos. Os próximos Estudos de Casos são sobre o CD. Neles são estudadas as estruturas atuais do CD e sugeridas formas de redundância, visando melhoria da disponibilidade do sistema estudado.

A Seção 7.1 apresenta os dados preliminares dos estudos. A infraestrutura inicial do ambiente estudado é apresentada na Subseção 7.1.1; a especificação do monitoramento é exposta na Subseção 7.1.2; assim como a realização dos cálculos de disponibilidade, tempos médios de falhas e reparos das camadas do sistema na Subseção 7.1.3. A Seção 7.2 apresenta o primeiro Estudo de Caso, que visa identificar a estrutura inicial e a disponibilidade do ambiente classificado como *baseline*. Na Subseção 7.2.1 são apresentados os modelos hierárquicos de disponibilidade em RBD e SPN utilizados como ferramenta para analisar a disponibilidade do sistema. A análise de sensibilidade da *baseline* é apresentada na Subseção 7.2.2. O segundo Estudo de Caso, apresentado na Seção 7.3, propõe alterações no ambiente e avalia o impacto destas modificações na disponibilidade do ambiente. As modificações sugeridas são guiadas pelo estudo de sensibilidade, aplicando redundâncias nas camadas de virtualização. O terceiro Estudo de Caso, apresentado na Seção 7.4, continua apresentando propostas de melhorias na infraestrutura interna do CD. Nele, é apresentada proposta de evolução do ambiente, contemplando um cenário com a redundância da camada de banco de dados. Para identificar o impacto desta sugestão, também é calculada a disponibilidade e verificada a necessidade de progressão no estudo. Na Seção 7.5, é realizado um comparativo das melhorias alcançadas nos primeiros Estudos de Casos, que visam aprimoramento interno ao CD.

Nas seções seguintes, são apresentados estudos que visam o aprimoramento do CD, identificando e sugerindo estruturas de redundância do CD. A Seção 7.6 apresenta o quarto Estudo de Caso que visa analisar o ambiente do CD e identificar pontos possíveis de aprimoramento. Para atingir o objetivo, foram aplicados dois níveis de simplificações aos modelos anteriormente apresentados. Finalmente, no quinto Estudo de Caso, exposto na Seção 7.7, é apresentada uma análise referente a redundância do CD e o impacto que trará na disponibilidade do sistema estudado.

7.1 ESTUDO DE CASO PRELIMINAR

Esta seção apresenta informações preliminares aos Estudos de Casos, expondo a infraestrutura inicial do sistema, denominada como *baseline*, monitoramento do sistema e os cálculos de disponibilidade, incluindo os tempos médios de falhas e reparo das camadas do sistema. Estas etapas são referentes às Macro-atividades 1, 2 e 3, respectivamente, apresentadas na Subseção 4.1.1.

7.1.1 Infraestrutura

O cenário de *baseline* do estudo está apresentado na Figura 32, e consiste nas camadas de conectividade, armazenamento, virtualização, orquestração, balanceador de carga, aplicação e banco de dados. Este estudo equivale à Macro-atividade 1, exposto na Subseção (4.1.1.1)

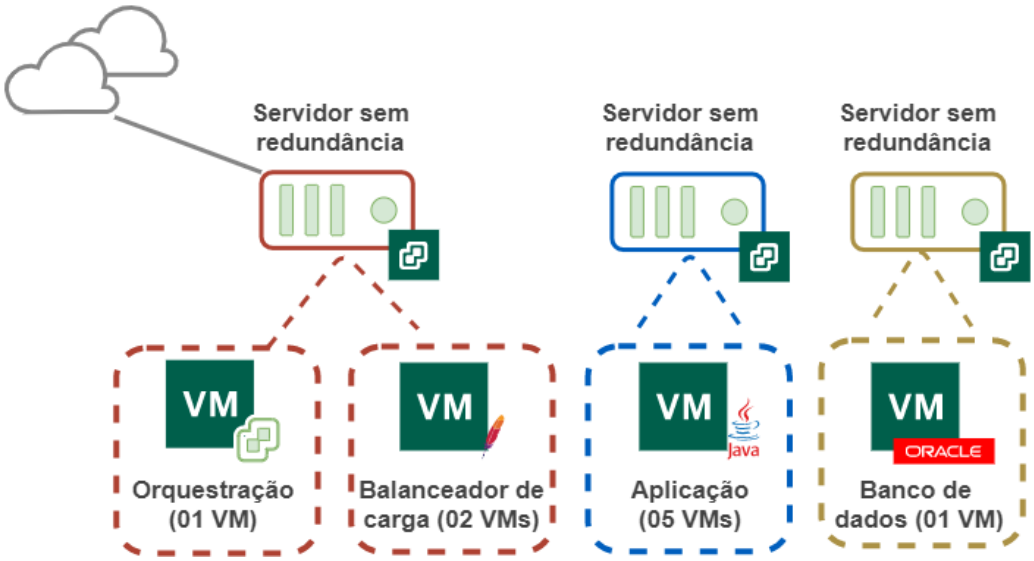


Figura 32 – Infraestrutura da *baseline*.

Algumas destas camadas possuem redundância e outras não. As camadas de armazenamento, virtualização, orquestração e banco de dados não possuem redundância de seus componentes. Por outro lado, as camadas SAN, conectividade e balanceador de carga possuem redundância e operam no modelo ativo/ativo com dois componentes cada uma delas. Já a camada de aplicação também possui redundância no modo ativo/ativo, porém com cinco componentes.

Como já estudado na Seção 5.2, a camada de conectividade é diretamente responsável pela disponibilidade do sistema por ser a conexão do ambiente com o mundo externo. Por este motivo, ao apresentar falha, o serviço se torna imediatamente inacessível, mesmo que internamente ele esteja no estado funcional. Já a camada de armazenamento também possui uma relação direta com a disponibilidade do ambiente, porém, atua de forma distinta. Caso apresente falha nesta camada, o sistema se torna indisponível sendo automaticamente desligado. O desligamento ocorre porque a estrutura lógica está guardada na unidade de armazenamento. Assim, após a camada de armazenamento voltar à normalidade, as máquinas virtuais passarão por todo o processo de inicialização. A indisponibilidade da camada de armazenamento também pode ocorrer na conexão do servidor físico com a unidade de armazenamento. Neste caso, as VMs hospedadas nesse servidor entrarão em um estado de falha e as máquinas virtuais nele hospedadas estarão indisponíveis. Se a falha ocorrer com servidores operando com mecanismos de redundância, pode ou não gerar um estado de indisponibilidade para o sistema.

Neste estudo, as hospedagens das VMs em servidores físicos são identificadas pela cor do seu contorno na imagem, informando que a VM utiliza os recursos físicos *memória, processamento, rede, etc.* daquele servidor ou grupo de servidores que possuam a mesma cor do contorno. Desta forma, as VMs não podem ser iniciadas em outro grupo de servidores físicos que não possuam a mesma cor do contorno. As camadas virtuais estão em estado operacional se um ou mais servidores desse grupo estiverem no mesmo estado. Exemplificando, na Figura 32 a camada de aplicação composta por cinco servidores virtuais, utiliza os recursos físicos do servidor de aplicação com a *mesma cor de contorno*. Se este servidor apresentar falha, todas as máquinas virtuais entrarão no estado de falha por interferência externa e não poderão ser iniciadas em outro servidor ou grupo de servidores físicos. A mesma ideia ocorre para todo o ambiente.

7.1.2 Monitoramento do sistema

Para aproximar os resultados do modelo com um sistema real, foi realizado o monitorando do sistema por um período de seis meses. Este estudo é referente à Macro-atividade 2 (4.1.1.2). A ferramenta escolhida para esta atividade foi o Zabbix[1]. No monitoramento, cada componente da camada virtual foi analisado individualmente por técnicas específicas que variavam conforme as características de cada um. Foi realizado o monitoramento de conectividade de rede para o sistema operacional; acessibilidade dos protocolos HTTP e HTTPS para balanceador de carga e aplicação; e, funções de monitoramento específicas para o banco de dados.

Os valores apresentados pelos fabricantes para equipamentos da estrutura computacional geralmente possuem um tempo de falha muito alto (SCHEER; DOLEZILEK, 2000; EXTREME, 2022). Em muitos casos, o tempo de falha é tão alto que a tecnologia se torna obsoleta antes de atingir o tempo informado pelo fabricante. Isto ocorre pelo fato de utilizarem uma taxa constante de falha para os equipamentos, porém, esta informação não considera o desgaste natural dos equipamentos (SCHEER; DOLEZILEK, 2000; SMITH, 2021; MACIEL, 2016). Visando uma melhor confiabilidade nos valores encontrados, se utilizou uma abordagem exemplificada em (SMITH, 2021) que se divide o valor encontrado na literatura dependendo do nível de confiabilidade e qual a fonte encontrada. Assim, os valores do MTTF referentes aos servidores físicos, switches SAN, switches ethernet e unidade de armazenamento foram retirados da literatura e os valores do MTTR foram retirados do ambiente estudado. Essa distinção foi necessária devido à existência de um contrato de manutenção que a instituição possui com o fabricante de alguns equipamentos. O contrato tem um SLA para troca de peças defeituosas em 48 horas para os switches SAN e armazenamento e 24 horas para os switches ethernet e servidor físico.

No monitoramento utilizado neste estudo, foram identificados apenas os momentos de falhas e reparos de cada componente. Sempre que houve uma alteração no estado do componente, foram coletadas data e hora para análise. Desta forma, se tornou possível identificar e isolar os TTF e Tempo de Reparo (TTR) de cada componente, permitindo a identificação de MTTF e MTTR através das Equações 2.9 e 2.10 respectivamente. Os valores obtidos através do monitoramento estão expostos no Apêndice A. Os valores obtidos de MTTF e MTTR obtidos através do monitoramento e da literatura são mostrados na Tabela 10. Esta mesma tabela apresenta, na quarta coluna, os valores da disponibilidade (A) de cada componente. Os

[1] https://www.zabbix.com

cálculos utilizados para descobrir esta disponibilidade, estão apresentados na Subseção 7.1.3.

Tabela 10 – Valores obtidos no monitoramento.

Componentes	**MTTF(h)**	**MTTR(h)**	**A**
Switch Ethernet	87600	24	0,999726
Switch SAN	87600	48	0,999452
Unidade de armazenamento	131400	4	0,999969
Hardware do servidor físico	26280	24	0,999087
Hipervisor (HP)	8760	1,67	0,999806
VM do orquestrador	56543	8,6	0,999848
VM do balanceador	98,11	0,12	0,998744
VM da aplicação	338,54	0,13	0,999612
VM do banco de dados	99,84	0,05	0,999489
Serviço de orquestração	46598	7,5	0,999839
Serviço do balanceador	1033,46	0,43	0,999575
Serviço da aplicação	34,71	0,13	0,996144
Serviço do banco de dados	271,43	0,91	0,996636

7.1.3 Cálculos da disponibilidade

Esta subseção apresenta os cálculos utilizados para encontrar os valores da disponibilidade, MTTF e MTTR das camadas, assim como a disponibilidade total do sistema estudado. Esta etapa foi realizada na Macro-atividade 3 (4.1.1.3). Pode-se comparar os valores obtidos nos cálculos com os modelos em RBD, para componentes em série, e a disponibilidade total do sistema encontrada pelo modelo em SPN. O intuito principal é apresentar como realizar os cálculos e validar os modelos concebidos.

O primeiro cálculo realizado identificou a disponibilidade das camadas do sistema, se utilizando dos valores do MTTF e MTTR de cada componente apresentado na Tabela 10. Em posse dos valores da disponibilidade, pode-se encontrar os MTTFs e MTTRs dos componentes que funcionam em série. Não serão realizados os cálculos dos MTTF e MTTR dos componentes em paralelo, pois estes serão modelados diretamente em SPN, porém, podem ser encontrados utilizando as Equações 2.7 e 2.11, respectivamente. Após encontrarmos a disponibilidade de todas as camadas do sistema, é possível realizar o cálculo da disponibilidade total do sistema,

que será utilizada no momento de validação do modelo apresentado.

Como apresentado na Seção 2.3, a disponibilidade de um equipamento é representada pela Equação 2.2. Para o cálculo da disponibilidade de equipamentos em paralelo, utilizou-se a Equação 2.17 e para cálculos de disponibilidade de equipamentos em série, a Equação 2.16. Em posse do valor da disponibilidade, pôde-se encontrar o MTTF dos equipamentos em série utilizando a Equação 2.5 e o MTTR utilizando a Equação 2.11.

Os primeiros cálculos a serem expostos são da estrutura computacional. Como pode-se perceber na Figura 14, exposta na Página 71, a camada de conectividade é composta por dois *switches ethernet* atuando no modo de ativo/ativo. Já a Camada de armazenamento é composta por três componentes, dois *switches SAN*, atuando em ativo/ativo e a unidade de armazenamento que funciona sem redundância. A camada de virtualização, mesmo sendo um componente da DCS, será calculada em conjunto com a estrutura lógica, por possuir uma relação direta entre elas.

Neste trabalho, para facilitar o entendimento e modelagem, a infraestrutura foi segmentada em camadas conforme a afinidade dos serviços ofertados. Desta forma, os cálculos seguiram a mesma estrutura.

Camada de conectividade:

Os dos *switches ethernet* operam de forma ativo/ativo, então, primeiro deve-se encontrar a disponibilidade de um equipamento e depois do conjunto de equipamentos (em paralelo). Na Equação 7.1, foi encontrado a disponibilidade individual de cada *Switch Ethernet* ($\boldsymbol{A_{sw}}$).

$$\begin{aligned} A_{sw} &= \frac{MTTF_{sw}}{MTTF_{sw} \ + \ MTTR_{sw}} \\ &= \frac{87600 \ h}{87600 \ h \ + \ 24 \ h} \\ A_{sw} &= 0,999726102 \ . \end{aligned} \tag{7.1}$$

Após encontrar a disponibilidade do *Switch Ethernet* foi possível calcular a disponibilidade da camada de conetividade, chamada na equação de $\boldsymbol{A_{net}}$. A Equação 7.2 é utilizada para descobrir a métrica desejada. De acordo com os cálculos realizados, o valor da disponibilidade da camada de conectividade é de $A_{net} = 0,999999925$.

$$
\begin{aligned}
A_{net} &= 1 - ((1 - A_{sw01}) \times (1 - A_{sw02})) \\
&= 1 - ((1 - 0,999726102) \times (1 - 0,999726102)) \\
A_{net} &= 0,999999925 \ .
\end{aligned} \tag{7.2}
$$

Por se tratar de uma camada atuando em paralelo, teve seu modelo diretamente em SPN. Sendo assim, não se fez necessário a realização dos cálculos para descobrir o MTTF e MTTR da camada. Os valores de MTTF e MTTR do componente são usados diretamente na modelagem em SPN. Esta abordagem foi utilizada, pois, caso realizasse a modelagem em RBD dos componentes em paralelo, a estrutura não contemplaria a capacidade de reparo dos equipamentos, apresentando uma confiabilidade sem utilização de reparo. Cenário que foge da realidade do ambiente estudado. Mesmo assim, a título de conhecimento, o cálculo do MTTF é apresentado para esta camada, utilizando a Equação 2.7, assim temos:

$$
\begin{aligned}
MTTF_{net} &= \frac{1}{\gamma_{net}} \times \sum_{i=1}^{n} \times \frac{1}{i} \\
&= \frac{1}{0,000011} \times \left(1 + \frac{1}{2}\right) \\
MTTF_{net} &= 131400 \ h \ ,
\end{aligned} \tag{7.3}
$$

e a descoberta do MTTR utilizando a Equação 2.11, então temos:

$$
\begin{aligned}
MTTR_{net} &= MTTF_{net} \times \left(\frac{UA_{net}}{A_{net}}\right) \\
&= 131400 \times \left(\frac{1 - 0,999999925}{0,999999925}\right) \\
MTTR_{net} &= 0,283775837 \ h \ .
\end{aligned} \tag{7.4}
$$

Camada de armazenamento:

Esta camada é composta por três componentes. Sendo dois *switches SAN* operando no modo ativo/ativo e uma unidade de armazenamento. Esta camada também foi modelada diretamente no SPN e não terá o cálculo dos valores do MTTF e MTTR.

Para calcular a disponibilidade desta camada também é necessário segmentá-la. Primeiro ocorre o cálculo da disponibilidade de um *swtitch SAN* ($\boldsymbol{A_{san}}$) e depois o valor da disponibilidade do conjunto de *switches*, chamada de $\boldsymbol{A_{sant}}$. O próximo passo é calcular a disponibilidade

da unidade de armazenamento, chamada de $\boldsymbol{A_{disk}}$. O último processo é o cálculo da disponibilidade total da camada de armazenamento, denominada de $\boldsymbol{A_{stg}}$.

- *Swtich SAN:* Por se tratar de equipamentos operando em ativo/ativo, pode-se utilizar do mesmo conceito apresentado na camada de conectividade. Desta forma, a Equação 7.5 calcula a disponibilidade individual do $\boldsymbol{A_{san}}$.

$$\begin{aligned} A_{san} &= \frac{MTTF_{san}}{MTTF_{san} \ + \ MTTR_{san}} \\ &= \frac{87600 \ h}{87600 \ h \ + \ 48 \ h} \\ A_{san} &= 0,999452 \ . \end{aligned} \tag{7.5}$$

Após encontrar a disponibilidade individual, pode-se calcular a disponibilidade total do conjunto de *switches* ($\boldsymbol{A_{sant}}$), utilizando a Equação 7.6. O valor da disponibilidade do conjunto de *switches SAN* foi $A_{sant} = 0,999999700$.

$$\begin{aligned} A_{sant} &= 1 - ((1 - A_{san01}) \ \times \ (1 - A_{san02})) \\ &= 1 - ((1 - 0,999452) \ \times \ (1 - 0,999452)) \\ A_{sant} &= 0,999999700 \ . \end{aligned} \tag{7.6}$$

- *Unidade de armazenamento:* A unidade de armazenamento é um equipamento voltado para segurança da informação e disponibilidade de serviços. O desenvolvimento deste tipo de equipamento é completamente estruturado, contemplando redundâncias de componentes físicos e lógicos. A Equação 7.7 mostra a disponibilidade da unidade de armazenamento, classificada por $\boldsymbol{A_{disk}}$.

$$\begin{aligned} A_{disk} &= \frac{MTTF_{disk}}{MTTF_{disk} \ + \ MTTR_{disk}} \\ &= \frac{131400 \ h}{131400 \ h \ + \ 4 \ h} \\ A_{disk} &= 0,9999696 \ . \end{aligned} \tag{7.7}$$

Após encontrar a disponibilidade do conjunto de *switches SAN* e da unidade de armazenamento, pode-se calcular a disponibilidade da camada de armazenamento ($\boldsymbol{A_{stg}}$). Esta

operação está exposta na Equação 7.8. Existe uma correlação entre os componentes, desta forma, caso um dos componentes apresente falha, toda a camada estará em falha. Assim, foi utilizado a equação da disponibilidade em série para encontrar o valor de $A_{stg} = 0,99996926$ representando a disponibilidade da camada.

$$\begin{aligned} A_{stg} &= A_{sant} \times A_{disk} \\ &= 0,999999700 \times 0,9999696 \\ A_{stg} &= 0,99996930 \ . \end{aligned} \tag{7.8}$$

Camada de virtualização:

A camada de virtualização é composta pelos *hardware* do servidor físico (HW) e o seu sistema operacional instalado nele classificado como hipervisor (HP), conforme já explicado na Seção 5.2. Esta camada é a aglutinação destes dois componentes. Os cálculos para encontrar a disponibilidade seguem o mesmo padrão utilizado até o momento. Inicialmente se encontra a disponibilidade do *hardware* do servidor físico ($\boldsymbol{A_{hw}}$), apresentado na Equação 7.9.

$$\begin{aligned} A_{hw} &= \frac{MTTF_{hw}}{MTTF_{hw} + MTTR_{hw}} \\ &= \frac{26280\ h}{26280\ h + 24\ h} \\ A_{hw} &= 0,999087591 \ . \end{aligned} \tag{7.9}$$

Depois encontra-se a disponibilidade do hipervisor ($\boldsymbol{A_{hp}}$), apresentada na Equação 7.10.

$$\begin{aligned} A_{hp} &= \frac{MTTF_{hp}}{MTTF_{hp} + MTTR_{hp}} \\ &= \frac{8760\ h}{8760\ h + 1.67\ h} \\ A_{hp} &= 0,99980675 \ . \end{aligned} \tag{7.10}$$

Sabendo das disponibilidades individuais, pode-se encontrar a disponibilidade da camada de virtualização $\boldsymbol{A_{srv}}$, apresentada na Equação 7.11.

$$
\begin{aligned}
A_{srv} &= A_{hw} \times A_{hp} \\
&= 0,999087591 \times 0,99980675 \\
A_{srv} &= 0,998894518 \ .
\end{aligned}
\tag{7.11}
$$

Por ser uma camada que opera em série, será realizado o cálculo do MTTF e do MTTR da camada. Estes valores serão conferidos com os valores obtidos no modelo de disponibilidade, apresentado na Seção 7.2.

A descoberta do MTTF se dá através da Equação 2.5, sendo a taxa inversamete proporcional ao MTTF, Equação 2.4. Desta forma, a Equação 7.12 representa o cálculo do MTTF da camada de virtualização.

$$
\begin{aligned}
MTTF_{srv} &= \frac{1}{\sum_{i=1}^{n} \gamma_i} \\
&= \frac{1}{(\gamma_{hw} + \gamma_{hp})} \\
&= \frac{1}{(0,000038 + 0,000116)} \\
MTTF_{srv} &= 6502,268041 \ h \ .
\end{aligned}
\tag{7.12}
$$

Em posse do valor do MTTF e da disponibilidade, pode-se encontrar o valor do MTTR através da Equação 2.11. Assim, a Equação 7.13 representa o cálculo do MTTR da camada de virtualização.

$$
\begin{aligned}
MTTR_{srv} &= MTTF_{srv} \times \left(\frac{UA_{srv}}{A_{srv}}\right) \\
&= 6502,268041 \times \left(\frac{0,001105482}{0,998894518}\right) \\
MTTR_{srv} &= 7,19609622 \ h
\end{aligned}
\tag{7.13}
$$

Camada de balanceador de carga:

A partir deste momento são apresentadas apenas as estruturas dos cálculos com os resultados obtidos. Outra característica adotada a partir deste ponto é que todos os componentes que possuam redundâncias atuando em ativo/ativo, serão duplicados. Desta forma, possuirão a mesma taxa γ, sendo a quantidade de componentes distintas de camada por camada.

A camada de balanceador também possui em seu primeiro nível hierárquico uma relação de série, tendo assim, apresentado os cálculos do MTTF, MTTR, e da disponibilidade. Esta camada é composta pelo SO e o *software* responsável pelo serviço, como explicamos na Seção 5.3. A expressão $\boldsymbol{A_{so(bal)}}$ se referencia a disponibilidade da máquina virtual e a expressão $\boldsymbol{A_{ap(bal)}}$ é a disponibilidade do apache.

A disponibilidade do SO do balanceador é encontrado em

$$A_{so(bal)} = \frac{MTTF_{so(bal)}}{MTTF_{so(bal)} \ + \ MTTR_{so(bal)}} = 0,998743686 \ , \tag{7.14}$$

e a disponibilidade da aplicação do balanceador de carga em

$$A_{ap(bal)} = \frac{MTTF_{ap(bal)}}{MTTF_{ap(bal)} \ + \ MTTR_{ap(bal)}} = 0,999575504 \ . \tag{7.15}$$

Estes dois componentes também possuem uma correlação, assim utilizo novamente a equação em série, como pode ser verificado na Equação 7.16.

$$A_{Bal01} = A_{so(bal)} \times A_{ap(bal)} = 0,998319724 \ . \tag{7.16}$$

Por também fazer parte do primeiro nível hierárquico da modelagem (RBD), são apresentados os cálculos de MTTF

$$MTTF_{bal(01)} = \frac{1}{(\gamma_{so(bal)} + \gamma_{ap(bal)})} = 89,60415853 \ h \ , \tag{7.17}$$

e do MTTR

$$MTTR_{bal(01)} = MTTF_{bal(01)} \times \left(\frac{UA_{bal(01)}}{A_{bal(01)}} \right) = 0,150813109 \ h \ . \tag{7.18}$$

Neste Estudo de Caso, como explicado na Subseção 7.1.1, esta camada possui duas VMs do balanceador de carga operando em ativo/ativo. Então, para calcular a disponibilidade total do serviço é preciso aplicar a equação da disponibilidade em paralelo, como exposto na Equação 7.19.

$$\begin{aligned} A_{balt} &= 1 - ((1 - A_{Bal01}) \ \times \ (1 - A_{Bal02})) \\ &= 1 - ((1 - 0,998319724) \ \times \ (1 - 0,998319724)) \\ A_{balt} &= 0,999997177 \ . \end{aligned} \tag{7.19}$$

Em posse do valor da disponibilidade total dos balanceadores de carga operando no modo ativo/ativo, pode-se identificar o valor da disponibilidade da camada de balanceador de carga (A_{Bal}). As máquinas virtuais estão hospedadas dentro de um servidor físico. Desta forma, para calcular a disponibilidade da camada em questão, deve-se aplicar novamente a equação em série utilizando os valores adquiridos na camada de virtualização e os balanceadores de carga, como exposto na Equação 7.20.

$$\begin{aligned} A_{bal} &= A_{srv} \times A_{balt} \\ &= 0,998894518 \times 0,999997177 \\ A_{bal} &= 0,998891698 \ . \end{aligned} \tag{7.20}$$

Camada de aplicação:

A camada de aplicação segue o mesmo raciocínio desenvolvido na camada de balanceador de carga. O primeiro objetivo é encontrar o valor da disponibilidade individual da aplicação, composta pela VM e aplicação. A expressão $\boldsymbol{A_{so(app)}}$, encontrada na Equação 7.21, se referencia a disponibilidade da VM.

$$A_{so(app)} = \frac{MTTF_{so(app)}}{MTTF_{so(app)} + MTTR_{so(app)}} = 0,999612463 \ , \tag{7.21}$$

e a expressão $\boldsymbol{A_{ap(tom)}}$, encontrada na Equação 7.22, representa a disponibilidade do Tomcat.

$$A_{ap(tom)} = \frac{MTTF_{ap(tom)}}{MTTF_{ap(tom)} + MTTR_{ap(tom)}} = 0,996143927 \ . \tag{7.22}$$

Sendo assim, a disponibilidade da aplicação é encontrada por

$$A_{app(01)} = A_{so(app)} \times A_{ap(tom)} = 0,995757885 \ . \tag{7.23}$$

Após encontrar o valor da disponibilidade da aplicação, pode-se encontrar os valores do MTTF em:

$$MTTF_{app(01)} = \frac{1}{(\gamma_{so(app)} + \gamma_{ap(tom)})} = 31,4824125 \ h \ , \tag{7.24}$$

e o valor do MTTR da aplicação em:

$$MTTR_{app(01)} = MTTF_{app(01)} \times \left(\frac{UA_{app(01)}}{A_{app(01)}} \right) = 0,134120984\ h\ . \tag{7.25}$$

Após saber o valor individual da disponibilidade da aplicação, pôde-se encontrar a disponibilidade da camada de aplicação $\boldsymbol{A_{appt}}$. Foi, ainda, replicada a quantidade de componentes da aplicação, sendo assim, eles possuem os mesmos valores da taxa γ. Foram utilizados cinco VMs operando em modo ativo/ativo, porém, para que o sistema esteja no estado operacional, se faz necessário que pelo menos uma VM esteja operando corretamente. Assim, foi utilizada a Equação 7.26.

$$\begin{aligned} A_{appt} &= 1 - ((1 - A_{app(01)}) \times (1 - A_{app(02)}) \times (1 - A_{app(03)}) \times (1 - A_{app(04)}) \times (1 - A_{app(05)})) \\ A_{appt} &= 0,999999999999\ . \end{aligned} \tag{7.26}$$

Foi utilizado um cenário bastante otimista para que o sistema estivesse funcional. Observou-se que apenas uma aplicação seria necessária para ofertar os serviços desejados. Porém, a título de conhecimento, são apresentados os cálculos para representar KooN. Neles, o sistema está no estado funcional caso existam 3 de 5, ou 4 de 5, ou 5 de 5 aplicações operacionais.

O Cálculo do KooN para 3 de 5 está representado na Equação 7.27.

$$\begin{aligned} A_{3oo5} &= \sum_{i=3}^{5} \binom{5}{3} A_{app(01)}^3 \times (1 - A_{app(01)})^2 \\ &= 20A_{app(01)}^3 \times (1 - A_{app(01)})^{5-3} + 5A_{app(01)}^4 \times (1 - A_{app(01)}) + A_{app(01)}^5 \\ A_{3oo5} &= 0,999999241\ . \end{aligned} \tag{7.27}$$

Caso o requisito fosse quatro aplicações no ar (4 de 5), seria representado pela Equação 7.28;

$$\begin{aligned} A_{4oo5} &= \sum_{i=4}^{5} \binom{5}{4} A_{app(01)}^4 \times (1 - A_{app(01)})^{5-4} \\ A_{4oo5} &= 5A_{app(01)}^4 \times (1 - A_{app(01)}) + 5A_{app(01)}^5 \\ A_{4oo5} &= 0,999821566\ . \end{aligned} \tag{7.28}$$

Porém, caso fosse necessário que todas as aplicações estivessem operacionais (5 de 5), representado na Equação 7.29.

$$\begin{aligned} A_{5oo5} &= \sum_{i=5}^{5} \binom{5}{5} A_{app(01)}^{5} \times (1 - A_{app(01)})^{5-5} \\ &= 5A_{app(01)}^{5} \\ A_{5oo5} &= 0,978968617 \ . \end{aligned} \tag{7.29}$$

Voltando aos parâmetros utilizados no trabalho, de possuir pelo menos uma aplicação no estado funcional, o cálculo da disponibilidade da camada de aplicação é apresentado. Da mesma forma que em todas as camadas virtuais, por estarem hospedadas no servidor físico, precisa-se utilizar a disponibilidade do servidor físico ($\boldsymbol{A_{srv}}$) com a da aplicação ($\boldsymbol{A_{app}}$), Equação 7.30.

$$\begin{aligned} A_{app} &= A_{srv} \ \times \ A_{appt} \\ &= 0,998894518 \ \times \ 0,999999999999 \\ A_{app} &= 0,998894518 \ . \end{aligned} \tag{7.30}$$

Camada de banco de dados:

A última camada virtual a calculada foi banco de dados. Ela segue o mesmo princípio das demais camadas de *software*, todavia, opera sem redundância. Desta forma, o valor obtido para a disponibilidade individual será utilizado como da camada de banco de dados.

Primeiramente é calculado o valor da disponibilidade da máquina virtual, representada na Equação 7.31 pela expressão $\boldsymbol{A_{so(db)}}$.

$$A_{so(db)} = \frac{MTTF_{so(db)}}{MTTF_{so(db)} \ + \ MTTR_{so(db)}} = 0,999489125 \ . \tag{7.31}$$

Em seguida se calcula a disponibilidade do Oracle, representada na Equação 7.32 por $\boldsymbol{A_{ap(db)}}$.

$$A_{ap(db)} = \frac{MTTF_{ap(db)}}{MTTF_{ap(db)} \ + \ MTTR_{ap(db)}} = 0,996636541 \ . \tag{7.32}$$

Em posse destes valores, pode-se calcular o valor da disponibilidade do banco de dados pela Equação 7.33.

$$A_{db(01)} = A_{so(db)} \times A_{ap(db)} = 0,996127385\ . \tag{7.33}$$

Os valores de MTTF e MTTR da camada estão expressos nas Equações 7.34 e 7.35 respectivamente.

$$MTTF_{db} = \frac{1}{(\gamma_{so(db)} + \gamma_{ap(db)})} = 72,99380448\ h \tag{7.34}$$

$$MTTR_{db} = MTTF_{db} \times \left(\frac{UA_{db}}{A_{db}}\right) = 0,283775837\ h\ . \tag{7.35}$$

Sabendo da disponibilidade dos componentes individualmente, pode-se calcular a disponibilidade da camada de banco de dados. Neste cálculo, como nos anteriores, a disponibilidade individual do banco foi aglutinada com a disponibilidade do servidor físico, expresso na Equação 7.36.

$$A_{db} = A_{srv}\ \times\ A_{db(01)} = 0,99502622\ . \tag{7.36}$$

Disponibilidade da *baseline*:

Após a descoberta da disponibilidade de cada camada do sistema, pode-se encontrar a disponibilidade da *baseline* ($\boldsymbol{A_{Baseline}}$). Como já apresentado na estrutura computacional, exposto na Figura 14, existe uma correlação entre as camadas. Todas as camadas precisam estar no estado funcional para que o sistema esteja operacional. Assim, também se utiliza a equação da disponibilidade em série para descobrir o valor da disponibilidade da *baseline*, conforme exposto na Equação 7.37.

$$\begin{aligned} A_{baseline} &= A_{net}\ \times\ A_{stg}\ \times\ A_{bal}\ \times\ A_{app}\ \times\ A_{db} \\ &= 0,999999925\ \times\ 0,999969\ \times\ 0,998892\ \times\ 0,998895\ \times\ 0,995026 \\ A_{baseline} &= 0,992794 \end{aligned} \tag{7.37}$$

A disponibilidade da *baseline* é de $A_{baseline} = 0,992794$ e o *downtime* anual de $63,12\ h$. O valor da disponibilidade será utilizado adiante, no momento de validar o modelo da *baseline*, na Subseção 7.2.1.

7.2 ESTUDO DE CASO I — AVALIAÇÃO DE DISPONIBILIDADE DA *BASELINE*

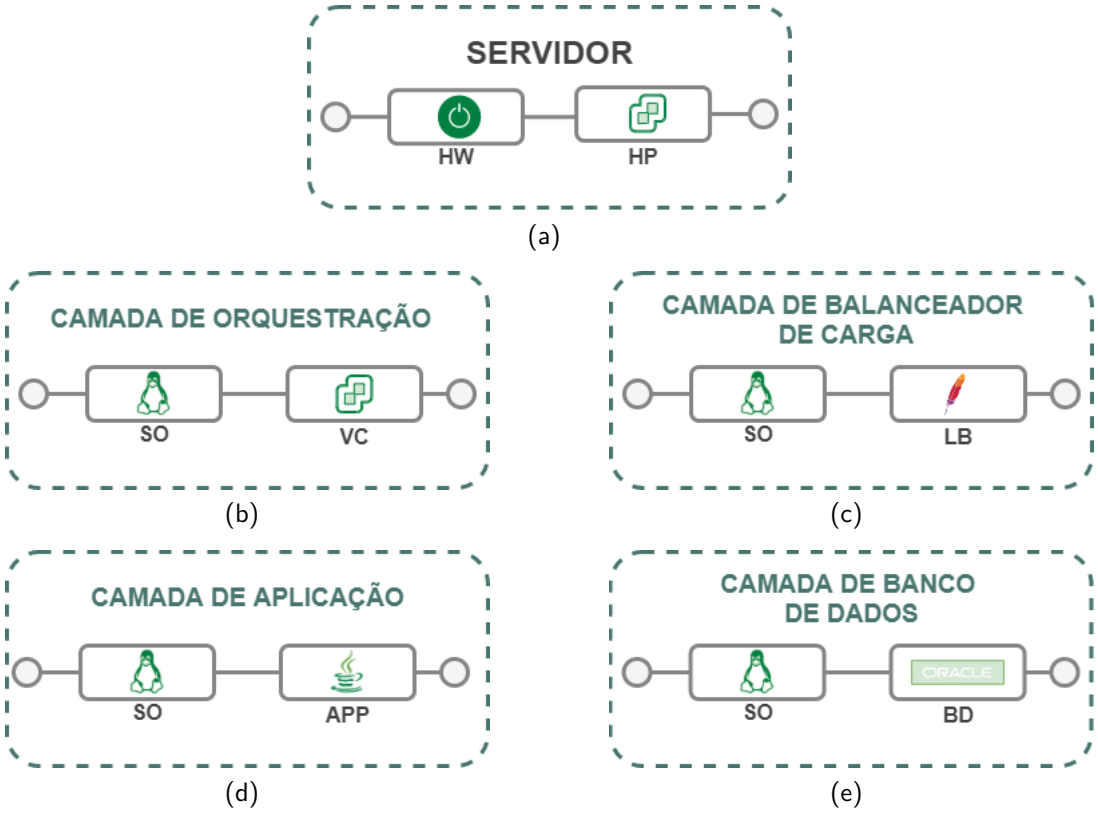
O objetivo principal deste Estudo de Caso é avaliar a disponibilidade do sistema estudado, identificando possíveis cenários que possibilitem uma melhora na métrica da disponibilidade. Para atingir a este objetivo, várias etapas precisam ser definidas e implementadas, permitindo uma avaliação de disponibilidade mais assertiva. Algumas delas estão expostas na Seção 7.1, tais como: análise do ambiente atual; classificado como *baseline*; implementação ou ajuste do monitoramento. Outras etapas estão apresentadas, como: concepção e validação do modelo que represente a *baseline*, na Subseção 7.2.1; e identificação de componentes sensíveis à disponibilidade, apresentada na Subseção 7.2.2. Estas etapas, equivalem às Macro-atividade 4, (Subseção 4.1.1.4) e Macro-atividade 5 (Subseção 4.1.2.1), respectivamente.

A primeira etapa deste estudo equivale à construção do modelo. Um modelo hierárquico de disponibilidade para a *baseline* foi concebido e com ele, calculada a métrica de disponibilidade do sistema. Em seguida, se criou um intervalo de confiança, utilizando a técnica de *boots-trapping*, visando identificar se os valores obtidos através dos cálculos e através do modelo de disponibilidade podem ser considerados aceitáveis matematicamente. Esta validação identifica se o modelo condiz com a realidade do ambiente estudado. O último passo identificou, através da análise de sensibilidade, os componentes mais sensíveis à disponibilidade do sistema.

7.2.1 Modelo de disponibilidade

Esta subseção apresenta o modelo de disponibilidade representando a *baseline* do ambiente estudado. Para a obtenção do modelo se fez necessário a construção de modelos hierárquicos, tendo o seu primeiro nível em modelos RBD e o segundo nível em modelos SPN. Esta estratégia possibilita a diminuição da complexidade do modelo em SPN, resultando em um modelo mais simples e que necessite de uma menor capacidade computacional. Porém, para elaboração deste modelo em SPN, se fez necessário calcular os valores de MTTF e MTTR dos componentes sintetizados em RBD e utilizá-los como insumo nos modelos em SPN. Os modelos em RBD sintetizam camadas e componentes específicas do ambiente. As camadas sintetizadas foram

a de virtualização, orquestração, balanceador de carga, aplicação e banco de dados. Estes modelos foram apresentados no Capítulo 6, tendo suas representações visuais nas Figuras 18 e 26. Para melhor entendimento, as figuras estão abaixo reapresentadas.



No modelo do servidor, foram sintetizados os componentes físicos (placa mãe, processador, memória...), classificado como (HW) e o sistema operacional nele instalado (hipervisor), classificado como (HP). Como resultado deste modelo em RBD, foram obtidos os valores de MTTF e MTTR da camada de virtualização. Os componentes virtuais também foram sintetizados no primeiro nível hierárquico em RBD. Todas as camadas virtuais são compostas pelo sistema operacional instalado na VM, classificado por (SO) e a aplicação instalada neste sistema operacional, que dependerá da camada virtual que está sendo modelada. A camada de orquestração, possui a aplicação VCenter, representada por (VC). A camada de balanceador de carga possui um apache com a função de balanceador de carga, representado por (LB). A camada de aplicação possui o Tomcat hospedando a aplicação, desta forma possui a representação (APP). Por último, a camada de banco de dados possui um Oracle e a representação (BD).

Como resultado dos modelos RBD, se obteve os valores de MTTF e MTTR das camadas sintetizadas. Estes resultados foram utilizados como insumos na representação destas mesmas

camadas no modelo em SPN. Os valores dos modelos RBD estão apresentados na Tabela 11. Pode-se perceber que os valores obtidos nos cálculos apresentados na Subseção 7.1.3 coincidem com os obtidos nos modelos em RBD, demonstrando que foram corretamente modelados. Os componentes ou camadas que não passaram por uma sintetização, foram modelados diretamente em SPN, assim, tiveram os seus valores de MTTF e MTTR diretamente adicionados no modelo em SPN.

Tabela 11 – Tempos de MTTF e MTTR utilizados no modelo em SPN - Estudo de Caso I.

Camadas/Componentes	**MTTF(h)**	**MTTR(h)**
Switch ethernet	87600	24
Switch SAN	87600	28
Unidade de armazenamento	131400	4
Camada de virtualização (Servidor)	6502,26	7,19
Camada de orquestração	25535	8
Camada de balanceador de carga	89,60	0,15
Camada de aplicação	31,48	0,13
Camada de banco de dados	72,99	0,28

Outra informação importante nos modelos em SPN são os tempos de inicialização de cada camada virtual. Estes se referem ao período necessário para que o serviço volte ao estado operacional após uma falha causada por interferências externas. Os tempos variam de acordo com cada serviço sendo aplicados nas transições de START_SFT de cada modelo. Os tempos estão expostos na Tabela 12.

Tabela 12 – Tempos de inicialização das camadas virtuais.

Transição	**Tempo de inicialização (h)**
START_VC	0,33
START_LB	0,08
START_APP	0,08
START_DB	0,16

O modelo de disponibilidade em SPN representando o Estudo de Caso I está exposto na Figura 33.

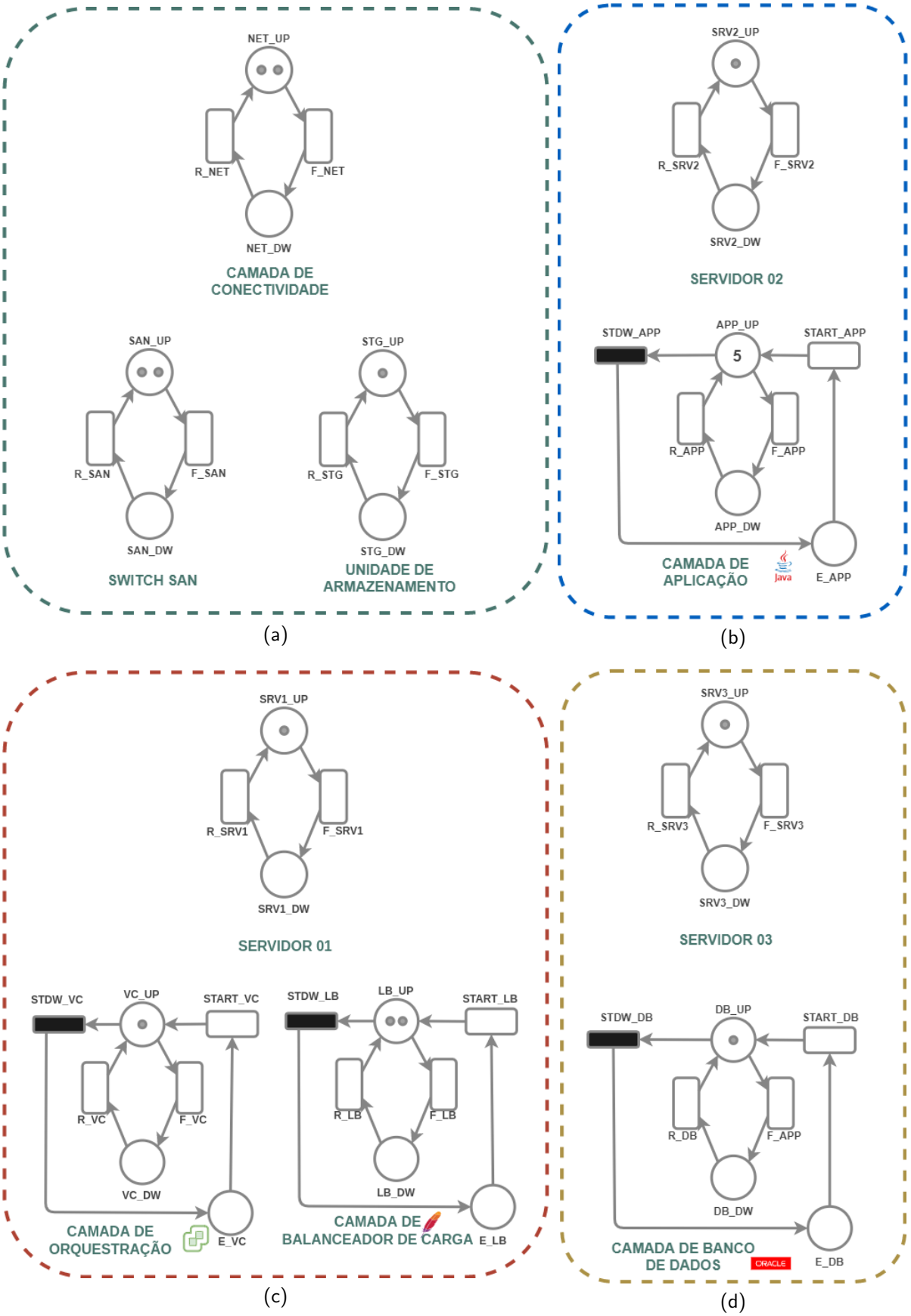


Figura 33 – Modelo em SPN representando o Estudo de Caso I.

Os componentes representando a DCS estão expostos na Figura 33a. Os modelos referentes as camadas de *software* estão expostos nas Figuras 33b, 33c e 33d, representando as camadas de aplicação, balanceador de carga e banco de dados, respectivamente. No *baseline* não existem servidores operando em redundância, desta forma, a modelagem dos servidores segue a estrutura apresentada na Figura 20, ilustrada na Página 82. Os componentes virtuais que não possuem configurações de redundância, camadas de orquestração e banco de dados, seguem a modelagem exposta na Figura 27a, apresentada na Página 90. Já as camadas que de possuem redundância, balanceador de carga e aplicação, utilizam a estrutura apresentada em na Figura 27b na Página 90.

Para que o sistema estudado esteja operacional, precisa-se ter pelo menos um componente de cada camada virtual no estado funcional (conectividade, armazenamento, virtualização, balanceador de carga, aplicação e banco de dados). A única camada que se relaciona com as camadas virtuais, mas não interfere diretamente na disponibilidade é a camada de orquestração. Assim, mesmo que ela esteja no estado de falha, o sistema pode estar operacional, pois ela possui uma atuação de monitoramento sobre as demais.

A disponibilidade é a probabilidade de que determinadas camadas estejam no estado operacional em simultâneo (BAUER; ADAMS, 2012), permitindo que o serviço ofertado esteja acessível e ofertando o serviço proposto. As expressões utilizadas na ferramenta Mercury responsáveis por determinar a probabilidade das camadas estarem operacionais estão expostas na Tabela 13.

Tabela 13 – Estudo de Caso I - Expressões para o cálculo da disponibilidade.

Métrica	**Expressão**
A_{net}	$P\{(\#NET\_UP > 0)\}$
A_{stg}	$P\{(\#STG\_UP > 0)$ and $(\#SAN\_UP > 0)\}$
A_{srv}	$P\{(\#SRV1\_UP > 0)\}$
A_{orq}	$P\{(\#VC\_UP > 0)\}$
A_{bal}	$P\{(\#LB\_UP > 0)\}$
A_{app}	$P\{(\#APP\_UP > 0)\}$
A_{db}	$P\{(\#DB\_UP > 0)\}$
A_t	$P\{(\#NET\_UP > 0)$ and $(\#LB\_UP > 0)$ and $(\#APP\_UP > 0)$ and $(\#DB\_UP > 0)\}$

Após realizar uma análise estacionária na ferramenta Mercury, obteve-se a disponibili-

dade mostrada na Tabela 14. Obteve-se o valor da disponibilidade referente a *baseline* de $\boldsymbol{A_t = 0,992698}$, possuindo um *downtime* anual de $\boldsymbol{63,96\,h}$.

Tabela 14 – Estudo de Caso I - Valores da disponibilidade

Métrica	Disponibilidade	#9's	dt anual(h)
A_{net}	0,999999	5,8908	0,00
A_{stg}	0,999969	4,5123	0,27
A_{srv}	0,998893	2,9559	9,70
A_{orq}	0,998498	2,8233	13,16
A_{bal}	0,998843	2,9369	10,13
A_{app}	0,998831	2,9323	10,24
A_{db}	0,994831	2,3955	45,24
A_t	0,992698	2,1365	63,96

Como explicado anteriormente, para que o sistema esteja no estado operacional, se faz necessário que no mínimo um componente de cada camada esteja no estado operacional. Para a camada de aplicação, que possui cinco componentes, essa proposta é bastante otimista. Desta forma, também é mostrado, para título de conhecimento, a fórmula utilizada na ferramenta Mercury para o KooN da aplicação. Esta fórmula considera que se faz necessário uma quantidade mínima da aplicação no estado funcional, para que todo o sistema esteja operacional. Exemplificando, caso seja realizado o cálculo da disponibilidade para o KooN de pelo menos três de cinco ($\boldsymbol{A_{3oo5}}$), quer dizer que o sistema será considerado operacional caso possua três, quatro ou cinco instâncias operacionais simultaneamente.

A expressão utilizada na descoberta do KooN, juntamente com o valor da disponibilidade para $\boldsymbol{A_{1oo5}}$ e $\boldsymbol{A_{5oo5}}$ estão apresentadas na Tabela 15. Na expressão $\boldsymbol{\#APP\_UP >= K}$, o termo $\boldsymbol{k}$ representa a quantidade mínima de aplicações que deve estar no estado operacional.

Por ser $\boldsymbol{k}$ a quantidade de instâncias operacionais, o valor da disponibilidade de $\boldsymbol{k = 1}$ é igual à disponibilidade encontrada anterioremente $\boldsymbol{A_t}$, apresentado na Tabela 14. Consequentemente a disponibilidade de $\boldsymbol{A_{5oo5}}$ será menor, já que todos as instâncias precisam estar operacionais para o sistema estar funcional. Mas, como explicado anteriormente, essas informações são a título de conhecimento, entretanto, a estrutura de apenas um componente é utilizada em cada camada para continuidade do trabalho.

Existem dois valores de disponibilidade que representam o ambiente estudado. Ambos possuem uma visão otimista e consideram o sistema no estado operacional quando no mínimo um

Tabela 15 – Estudo de Caso I - Expressão de disponibilidade - KooN

Métrica	Expressão	A	dt anual(h)
A_{KooN}	$P\{(\#NET\_UP > 0)$ and $(\#LB\_UP > 0)$ and $(\#APP\_UP >= \boldsymbol{k})$ and $(\#DB\_UP > 0)\}$	✗	✗
A_{1oo5}	$P\{(\#NET\_UP > 0)$ and $(\#LB\_UP > 0)$ and $(\#APP\_UP >= \mathbf{1})$ and $(\#DB\_UP > 0)\}$	0,992698	63,96
A_{5oo5}	$P\{(\#NET\_UP > 0)$ and $(\#LB\_UP > 0)$ and $(\#APP\_UP >= \mathbf{5})$ and $(\#DB\_UP > 0)\}$	0,971499	249,66

componente de cada camada está no estado operacional. O primeiro valor de disponibilidade foi obtido através dos cálculos preliminares e o segundo através da construção de um modelo representando o ambiente estudado. Precisa-se verificar se o modelo construído condiz com o ambiente real. Para isto, pode ser calculado o Intevalo de Confiança (IC). O intervalo de confiança identifica estatisticamente que o valor obtido no modelo não pode ser desconsiderado como válido (DEVORE, 2008).

Os valores de TTFs e TTRs dos componentes foram registrados a partir do monitoramento, ocasionando variações na quantidade de registros de cada componente, dependendo da estabilidade da camada monitorada. Os componentes mais estáveis apresentam poucas variações e, consequentemente, menor quantidade de registros no monitoramento. Isto ocorre pelo fato do monitoramento realizado identificar os momentos de falhas e reparos dos componentes, registrando data e hora de cada alteração de estado dos componentes. A camada de aplicação, por exemplo, representou a maior variação entre os componentes monitorados, registrando 140 alterações de estado, incluindo dados referentes ao SO e ao Tomcat. Por outro lado, a camada mais estável foi a camada do balanceador de carga, com 48 registros de alterações de estados. Estes valores estão expostos no Apêndice A.

Com essa característica do monitoramento, as informações coletadas foram suficientes para realizar cálculos e construir o modelo, porém, são insuficientes para gerar uma população estatisticamente aceitável e gerar um IC. Para suprir esta quantidade insuficiente de dados, optou-se pela técnica de *Bootstrapping* para calcular o IC. De acordo com (DIXON, 2006), *Bootstrapping* é um mecanismo de reamostragem capaz de gerar estatísticas populacionais por amostragem de um conjunto de dados. Como apresentado anteriormente, por não terem sido monitoradas, as camadas de conectividade e armazenamento não passaram pelo Bootstrapping, tendo seus valores de MTTF e MTTR foram retirados da literatura em conjunto com a

infraestrutura do sistema. Apenas as camadas (LB, APP e BD) passaram pelo Bootstrapping.

A técnica utilizada gera 1000 reamostras de TTF e TTR para cada componente, utilizando como base os dados obtidos no monitoramento. Todas as camadas virtuais são compostas por SO's e aplicações. Desta forma, para a camada de aplicação, por exemplo, foram gerados 1000 reamostras do TTF e TTR referentes ao SO, e 1000 reamostras de TTF e TTR referentes ao Tomcat. O próximo passo ordenou as reamostras para descobrir os percentis para o intervalo de confiança de 95%. Para isso, se utilizou as amostras 25º e 975º de TTF e TTR ((TTF_{25} e TTR_{25})) e ((TTF_{975} e TTR_{975})) e se realizou o cálculo da disponibilidade para cada um deles. Com estes valores, se obteve a disponibilidade $\boldsymbol{A_{2,5\%}}$ e $\boldsymbol{A_{97,5\%}}$ de cada componente das camadas virtuais. Estes valores são justamente o nosso intervalo de confiança. Assim, ainda utilizando a camada de aplicação como exemplo, se obteve a $\boldsymbol{A_{so(2,5\%)}}$ e a $\boldsymbol{A_{so(97,5\%)}}$, representando os valores do SO, e $\boldsymbol{A_{tom(2,5\%)}}$ e a $\boldsymbol{A_{tom(97,5\%)}}$, representando os valores do Tomcat.

Em seguida, foi realizado o cálculo da disponibilidade da camada de aplicação em 2,5% com a equação $\boldsymbol{A_{app(2,5\%)} = A_{so(2.5\%)} \times A_{tom(2.5\%)}}$, e para 97,5% com a equação $\boldsymbol{A_{app(97,5\%)} = A_{so(97,5\%)} \times A_{tom(97,5\%)}}$. O mesmo processo foi adotado para todas as camadas virtuais. Em posse dos valores das disponibilidades individuais, calculou-se o valor da disponibilidade para o sistema com o intervalo de confiança de 95%, $\boldsymbol{A_{t(2,5\%)}}$ e $\boldsymbol{A_{t(97,5\%)}}$.

Os valores da disponibilidade referentes ao IC de 95% são comparados com o valor da disponibilidade obtido no modelo SPN, permitindo, a partir deste momento, identificar se o modelo construído corresponde ao sistema real. Caso o valor da disponibilidade obtido no modelo SPN, esteja contido no intervalo de confiança gerado pelo *bootstrap*, pode-se dizer que o modelo foi validado. Esta comparação está exposta na Tabela 16, onde a expressão $\boldsymbol{A_{calc}}$ representa a disponibilidade calculada na Subseção 7.1.3, e a expressão $\boldsymbol{A_{modelo}}$ representa a disponibilidade descoberta no modelo de disponibilidade.

Tabela 16 – Estudo de Caso I - Intervalo de confiança.

Métrica	$\boldsymbol{A_{calc}}$	$\boldsymbol{A_{modelo}}$	**IC 95%**
A_t	0,992794	0,992698	$0{,}9892 < \theta < 0{,}9932$

Como verificado na Tabela 16, o intervalo de confiança contém a métrica de disponibilidade do modelo proposto. Portanto, não se pode refutar que o modelo não representa o sistema real.

O resultado apresentado na Tabela 14 mostra a disponibilidade do sistema estudado com pelo menos um componente de cada camada ativo. A disponibilidade total da *baseline* é $\boldsymbol{A_t = 0,992698}$, com um tempo de indisponibilidade anual em horas de $\boldsymbol{63,96\ h}$. A métrica da disponibilidade possui um valor baixo, resultando em um alto tempo de indisponibilidade anual em comparação com outros sistemas hospedados em nuvem. Visando aprimorar essa métrica, uma análise de sensibilidade foi realizada para identificar qual componente é mais sensível e possui o maior impacto quando realizado alterações no ambiente. Esta abordagem é fundamental para o objetivo de aprimorar a disponibilidade do sistema com mais precisão.

7.2.2 Análise de sensibilidade

A análise de sensibilidade (AS) identifica quais parâmetros possuem interferências mais significativas na disponibilidade ao serem variados. Primeiramente, experimentos individuais foram realizados, variando os valores de cada parâmetro. Em seguida, analisou-se o impacto na métrica estudada. Quanto maior o impacto, maior a capacidade de interferir na disponibilidade. Com este procedimento, identifica-se o parâmetro a ser ajustado para obter uma melhor disponibilidade para o ambiente estudado.

Para esta análise os experimentos foram realizados utilizando a ferramenta Mercury, assumindo-se uma variação de 50% para menos e 50% para mais dos valores de cada parâmetro. Estes valores foram apresentados previamente na Tabela 11. Dentro deste intervalo mínimo e máximo, foi configurada uma variação de 10% do valor entre os experimentos, totalizando 10 experimentos para cada parâmetro. O passo seguinte, conforme apresentado na Seção 2.4, foi a identificação do índice de sensibilidade para cada parâmetro, utilizando a Equação 2.15. Como exemplo há a apresentação do cálculo utilizado para a descoberta do índice de sensibilidade no parâmetro $\boldsymbol{MTTF_{db}}$,

$$
\begin{aligned}
S_{p(A)} &= \frac{A_{p(max)} - A_{p(min)}}{A_{p(min)}} \\
S_{MTTF_{db}(A)} &= \frac{A_{t(max)} - A_{t(min)}}{A_{t(min)}} \\
&= \frac{0,993782 - 0,988812}{0,988812} \\
S_{MTTF_{db}(A)} &= 0,005026234\ .
\end{aligned}
\tag{7.38}
$$

Este mesmo processo é realizado para todos os parâmetros, obtendo a classificação e identificação dos parâmetros que apresentaram as variações mais significativas em comparação com a métrica final. Nos dados apresentados na Tabela 17, o valor mais alto significa que aquele parâmetro possui o impacto mais significativo na disponibilidade final.

Tabela 17 – Análise de sensibilidade.

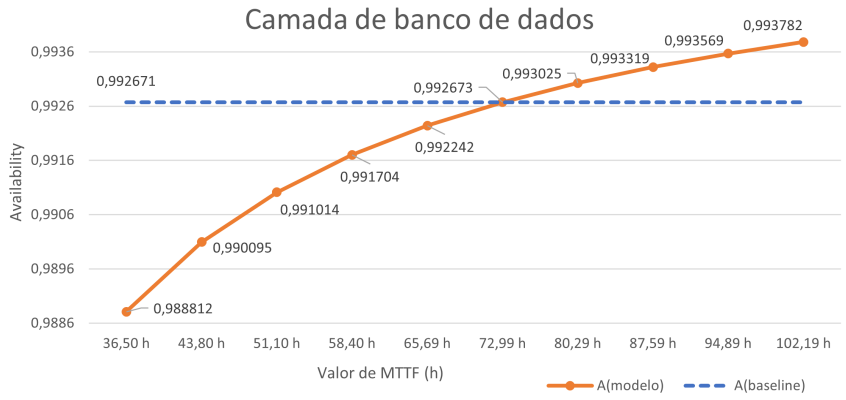
Parâmetro	**Índice**	**Parâmetro**	**Índice**
$MTTF_{db}$	0,005026	$MTTF_{lb}$	0,000017
$MTTF_{srv}$	0,004507	$MTTF_{app}$	0,000012
$MTTR_{db}$	0,003849	$MTTF_{san}$	0,000008
$MTTR_{srv}$	0,002996	$MTTR_{san}$	0,000007
$MTTF_{stg}$	0,000076	$MTTF_{lb}$	0,000006
$MTTF_{vc}$	0,000072	$MTTF_{net}$	0,000000
$MTTR_{stg}$	0,000055	$MTTR_{net}$	0,000000
$MTTF_{vc}$	0,000048	$MTTR_{app}$	0,000000

O ambiente estudado possui muitos parâmetros. Assim, os quatro parâmetros que obtiveram as maiores variações na métrica de disponibilidade, juntamente com os parâmetros que obtiveram as menores variações estão sendo apresentados. Visando melhor entendimento, uma análise percentual dos índices de sensibilidade foi realizada. Esta análise está exposta na Tabela 18 .

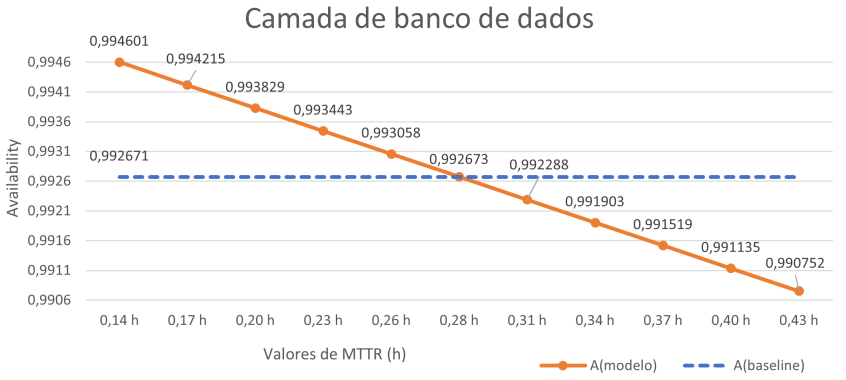
Tabela 18 – Variação percentual da análise de sensibilidade.

Parâmetro	**Variação percentual**	**Parâmetro**	**Variação percentual**
$MTTF_{db}$	50,26 %	$MTTR_{lb}$	0,06 %
$MTTF_{srv}$	45,07 %	$MTTF_{net}$	0,00 %
$MTTR_{db}$	38,85 %	$MTTR_{net}$	0,00 %
$MTTR_{srv}$	29,97 %	$MTTR_{app}$	0,00 %

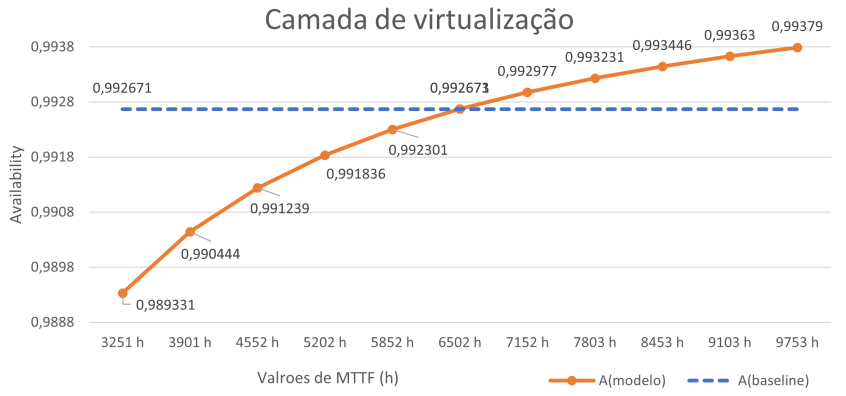
Analisando os dados expostos nas Tabelas 17 e 18, pôde-se detectar que os parâmetros da camada de banco de dados e virtualização apresentaram as maiores taxas e percentuais, ao contrário da camada de aplicação e conectividade que obtiveram os menores índices. Na apresentação visual, exposta na Figura 34, foi possível comparar os quatro maiores índices de sensibilidade com a disponibilidade da *baseline*.



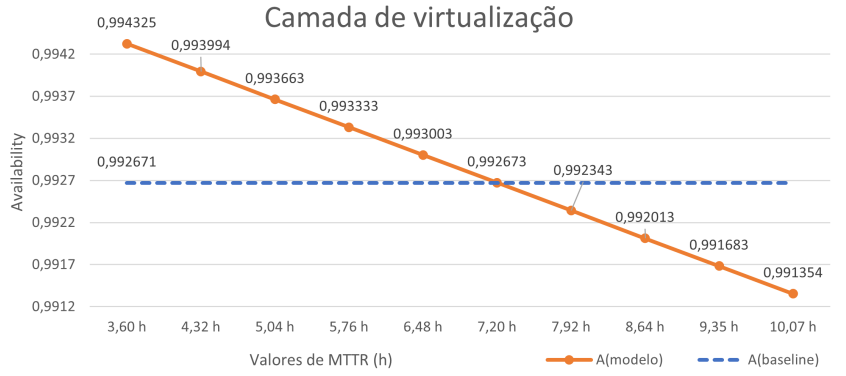
(a)



(b)



(c)



(d)

Figura 34 – Gráficos com os maiores índices na análise se sensibilidade.

Nos Gráficos 34a e 34b, são apresentados as variações obtidas nos experimentos para os valores de MTTF e MTTR da camada de banco de dados, respectivamente. Já nos Gráficos 34c e 34d, são expostos às variações do MTTF e MTTR para a camada de virtualização. Os gráficos apresentam as variações dos valores no eixo X e o valor de disponibilidade obtida no eixo Y. A linha azul representa a disponibilidade da *baseline*. Exemplificando, no Gráfico 34a, apresenta-se o experimento com o valor do MTTF_{db} . O primeiro experimento utiliza $\boldsymbol{36,5\ h}$ como valor do MTTF da camada de banco de dados e obteve uma disponibilidade de $\mathbf{0,988812}$, enquanto a disponibilidade da *baseline* é de $\mathbf{0,992698}$. No segundo experimento, foi aumentado em 10% o valor do MTTF, sendo utilizado $\boldsymbol{43,8\ h}$ que obteve uma disponibilidade de $\mathbf{0,990095}$. Assim foi procedido até o décimo experimento que utilizou $\boldsymbol{102,19\ h}$ como MTTF com o resultado de $\mathbf{0,993782}$ para a disponibilidade. Foi identificado que quanto maior o valor de MTTF_{db} , maior o valor de disponibilidade.

A mesma técnica foi usada para MTTR_{db} , exposta no Gráfico 34b. O menor valor foi de $\boldsymbol{0,14\ h}$ e obteve uma disponibilidade de $\mathbf{0,994601}$. A partir deste ponto, os valores do MTTR do banco de dados foram aumentados em 10% para cada experimento. Assim, o segundo experimento usou o valor de $\boldsymbol{0,17\ h}$ e obteve $\mathbf{0,994215}$ de disponibilidade. O último experimento foi com $\boldsymbol{0,4\ h}$ de MTTR_{db} , com o resultado de $\mathbf{0,991135}$. Analisando este conjunto de experimentos, percebe-se uma direção reversa. Quanto maior o valor de MTTR_{db}, menor o valor de disponibilidade.

O mesmo ocorre para a camada de virtualização. Os resultados obtidos para os componentes DB e SRV são compreensíveis. Quanto mais alto o valor do MTTF, maior o tempo de atividade do componente, resultando em melhor disponibilidade. Por outro lado, quanto maior o valor do MTTR, maior o tempo de falha, resultando em menor disponibilidade. No entanto, nem todos os componentes apresentam uma variação tão expressiva na disponibilidade para mudanças de parâmetros.

É exposto na Figura 35 os gráficos com os componentes que obtiveram as menores variações da disponibilidade, consequentemente menores taxas de sensibilidade. Pode-se perceber que a camada de conectividade, apresentada nos Gráficos 35a e 35b, juntamente com o parâmetro de MTTR da camada de aplicação, exposta no Gráfico 35c, não possuem variações no valor da disponibilidade ao terem seus parâmetros alterados. Estes parâmetros possuem naturalmente uma pequena melhora na disponibilidade, todavia, os seus valores já são tão satisfatórios que sua interferência na métrica estudada é mínima.

O último parâmetro apresentado visualmente é o MTTR da camada do balanceador de

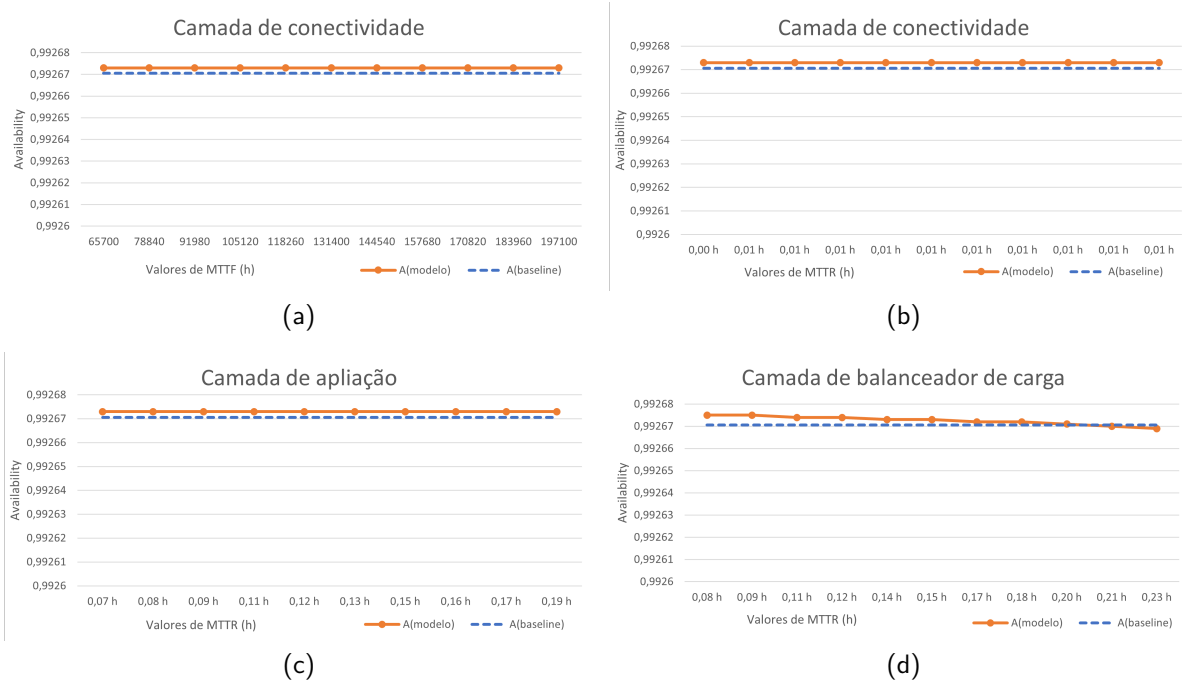


Figura 35 – Gráficos com os menores índices na análise se sensibilidade.

carga, exposto no Gráfico 35d. Este parâmetro possui um comportamento um pouco diferente em comparação aos demais deste grupo de gráficos. Ao realizar a variação do MTTR no experimento, a disponibilidade final sofre uma pequena alteração, como identificado na Tabela 18. Isto se explica pelo alto valor de disponibilidade da camada de balanceador de carga. Ao isolar e comparar apenas esta disponibilidade, pode-se perceber que ela já é bastante elevada em comparação com a *baseline*. Desta forma, seria necessária uma alteração muito grande no MTTR do balanceador para obter uma melhora na disponibilidade final do sistema.

Analisando os gráficos, pode-se perceber que as melhores disponibilidades foram alcançadas nos experimentos com os parâmetros dos MTTRs, embora tenham uma variação mais modesta que os MTTFs. Com essa análise de sensibilidade, é possível identificar quais parâmetros podem ser melhorados para alcançar um efeito mais assertivo no valor de disponibilidade do ambiente.

7.3 ESTUDO DE CASO II — APRIMORAMENTO DA DISPONIBILIDADE DO SISTEMA

O objetivo deste Estudo de Caso é aprimorar a métrica de disponibilidade do sistema estudado. Este estudo se refere as Macro-atividades 6 (Subseção 4.1.2.1), 7 (Subseção 4.1.2.3) e 8 (Subseção 4.1.2.4). Foram propostas alterações na infraestrutura dos componentes internos ao CD. Como identificado na análise de sensibilidade exposta na Subseção 7.2.2, as camadas de virtualização e de banco de dados possuem os parâmetros mais sensíveis à alteração na

disponibilidade. Sabendo disto, foi iniciado o Estudo de Caso aprimorando os parâmetros necessários. Na análise de sensibilidade, também foi identificado que as melhores disponibilidades foram obtidas aprimorando o MTTR dos componentes. A melhora do MTTR resulta em um maior tempo no estado operacional, pois o serviço sai do estado de falha com mais rapidez. Existem várias abordagens possíveis para conseguir este objetivo. A adotada neste estudo é a utilização de redundâncias nos serviços desejados.

A primeira sugestão de aprimoramento é duplicar a camada de virtualização. Um modelo em SPN com as modificações desejadas foi concebido, visando identificar o impacto desse processo na disponibilidade do ambiente. Neste primeiro momento, recorreu-se à configuração do servidor *cold standby* para as camadas de virtualização, responsáveis por hospedar as camadas de orquestração e balanceador de carga. Para as camadas de aplicação e banco de dados, foi utilizada a configuração do servidor em HA, conforme mostrado na Figura 36. Foram utilizadas formas de redundâncias distintas na configuração da camada de virtualização visando identificar a diferença de disponibilidade entre as elas. Como explicado anteriormente, foram escolhidas essas categorias de redundâncias pela capacidade de implementação destes cenários no ambiente real.

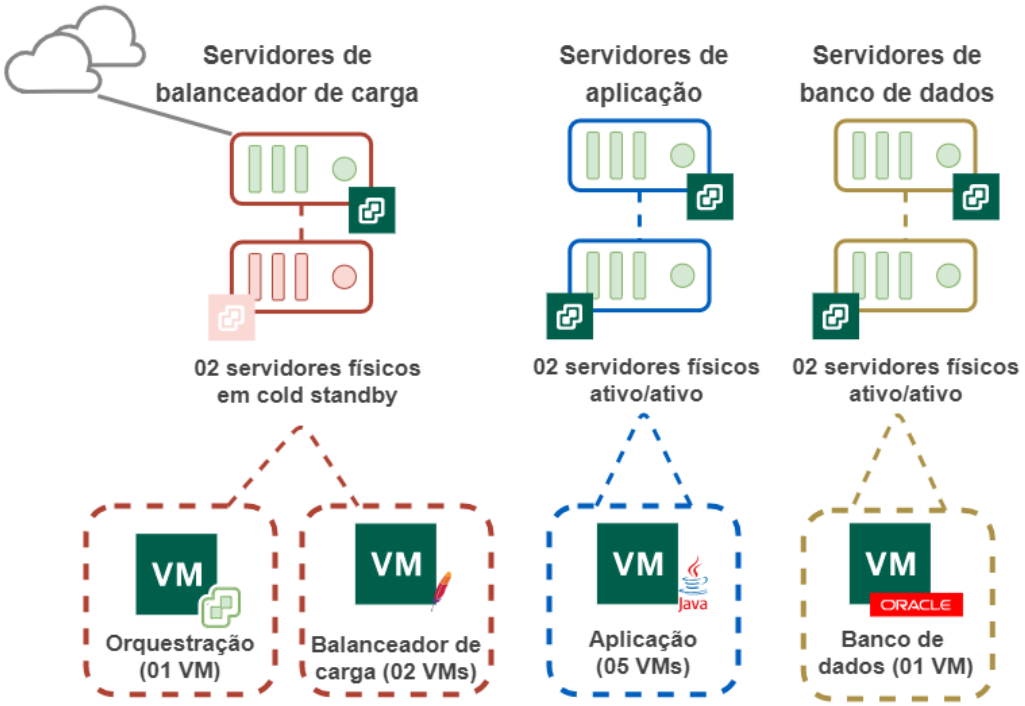


Figura 36 – Infraestrutura do Estudo de Caso II.

Foi abstraído da imagem os componentes da estrutura computacional, porém estes componentes estão contemplados na modelagem. As demais características de hospedagem das

VMs permanecem as mesmas. A definição de onde a VM está hospedada é dada pela cor do contorno. Como informado anteriormente, na Seção 5.2, utiliza-se a capacidade máxima de recurso de cada servidor físico. Por este motivo, não se consegue de imediato realizar a redundância da camada de banco de dados, que resultaria no aumento na quantidade de máquinas virtuais desta camada. Primeiramente é necessário o ajuste da quantidade de servidores. Para não alterar apenas a quantidade de servidores da camada de banco, foi ajustada a configuração para todo o ambiente.

7.3.1 Modelo de disponibilidade

Esta subseção expõe o modelo de disponibilidade representando um ambiente proposto. Os tempos de MTTF e MTTR utilizados como insumo no modelo em SPN estão expostos na Tabela 11. Também foi utilizada a ferramenta Mercury para realizar os cálculos de disponibilidade. A apresentação visual dos modelos está segmentada em figuras diferentes, porém, a modelagem continua unificada. A modelagem das camadas de conectividade e de armazenamento serão as mesmas apresentadas no Estudo de Caso I, exposto na Subseção 7.2.1. A modelagem representando a camada de orquestração e balanceador de carga estão expostos na Figura 37.

Nesta estrutura foi utilizada a configuração de servidores funcionando em *cold standby* e o funcionamento desta modelagem está apresentada na Subseção 6.1.2. Por utilizar esta configuração na camada de virtualização, se fez necessário adotar a modelagem da camada de *software* para servidores em *cold standby*, exposta na Subseção 6.1.2.

O modelo representando os dois servidores está apresentado na Figura 37a. Nele pode-se identificar o servidor principal pela nomenclatura SRV1 e o servidor secundário por SRV11. No funcionamento básico do modelo, apenas um dos dois servidores se encontra no estado operacional, entretanto, pode ocorrer de os dois estarem no estado operacional. Porém, este estado é temporário até realizar as migrações das máquinas virtuais. A Figura 37b representa a camada de orquestração e a Figura 37c ilustra a camada de balanceador de carga. Ambas camadas podem estar hospedadas nos dois servidores. Caso o token esteja em VC1_UP ou LB1_UP simboliza que a VM está hospedada no SRV1. Por outro lado, caso o token esteja em VC2_UP ou LB2_UP, as VMs estarão hospedadas no servidor secundário. As expressões do cálculo da disponibilidade estão apresentadas na Tabela 19.

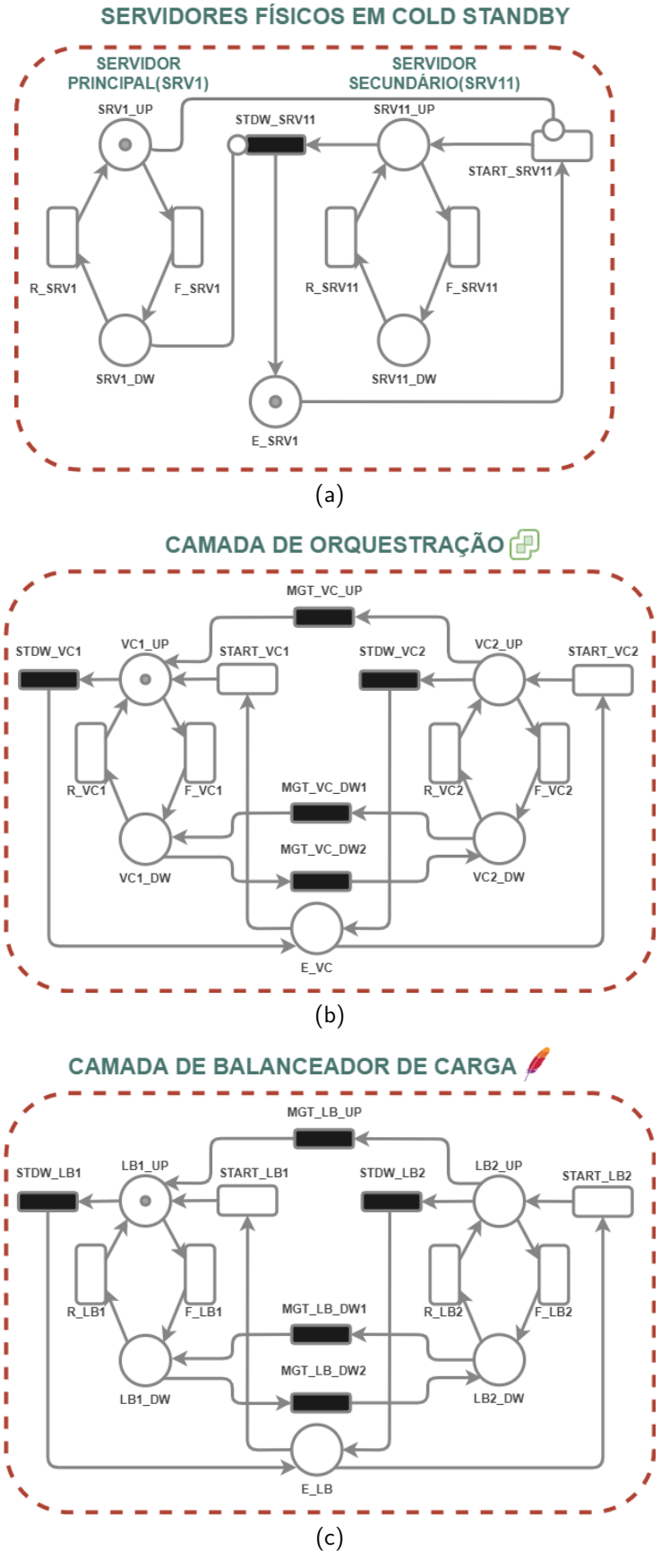


Figura 37 – Estudo de Caso II - Camadas de orquestração e balanceador de carga.

Tabela 19 – Estudo de Caso II - Expressões para o cálculo da disponibilidade das camadas de orquestração e balanceador de carga.

Métrica	Expressão
A_{srv}	$P\{((\#SRV1\_UP > 0) \text{ or } (\#SRV11\_UP > 0))\}$
A_{orq}	$P\{(((\#VC1\_UP) + (\#VC2\_UP)) > 0)\}$
A_{bal}	$P\{(((\#LB1\_UP) + (\#LB2\_UP)) > 0)\}$

A representação para as camadas de aplicação e banco de dados estão expostas na Figura 38. Cada uma das camadas utiliza a configuração de servidores em ativo/ativo para hospedar as suas respectivas VMs. A Figura 38a representa os servidores da camada de aplicação e a Figura 38b apresenta os servidores da camada de banco de dados. A ilustração da camada de aplicação está apresentada na Figura 38c e a da camada de banco de dados na Figura 38d.

As expressões do cálculo da disponibilidade estão apresentadas na Tabela 20.

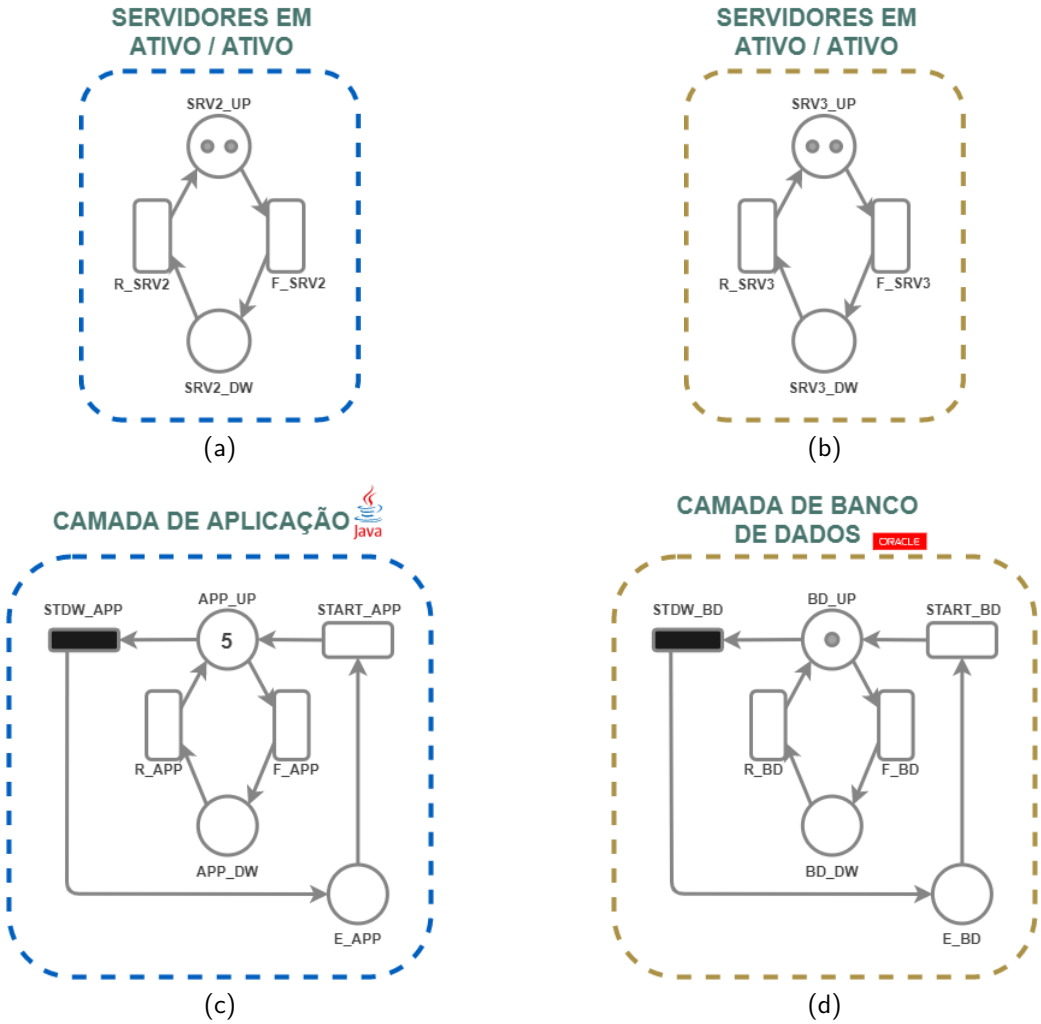


Figura 38 – Estudo de Caso II - Camadas de aplicação e banco de dados.

Tabela 20 – Estudo de Caso II - Expressões para o cálculo da disponibilidade das camadas de aplicação e banco de dados.

Métrica	Expressão
A_{srv2}	$P\{(\#SRV2\_UP > 0)\}$
A_{srv3}	$P\{(\#SRV3\_UP > 0)\}$
A_{app}	$P\{(\#APP\_UP \geq 1)\}$
A_{bd}	$P\{(\#BD\_UP \geq 1)\}$

A expressão utilizada para o cálculo da disponibilidade total do modelo está apresentada na Tabela 21.

Tabela 21 – Estudo de Caso II - Expressão de disponibilidade.

Métrica	Expressão
A_t	$P\{(\#NET\_UP > 0)$ and $(((\#LB1\_UP) + (\#LB2\_UP)) > 0)$ and $(\#APP\_UP > 0)$ and $(\#DB1\_UP > 0)\}$

Após realizar análise estacionária na ferramenta Mercury, obteve-se a disponibilidade mostrada na Tabela 22. Os valores das camadas de conectividade e armazenamento não estão expostos por não terem sofrido alterações em relação ao Estudo de Caso I (Tabela 14). O valor da disponibilidade referente ao segundo Estudo de Caso é: $\boldsymbol{A_t = 0,995931}$, possuindo um *downtime* anual de $\boldsymbol{35,64\,h}$.

Tabela 22 – Estudo de Caso II - Valores da disponibilidade.

Métrica	Disponibilidade	#9's	dt anual(h)
A_{srv1}	0,999973	4,5084	0,27
A_{srv2-3}	0,999998	5,1271	0,06
A_{orq}	0,999574	3,3704	3,73
A_{bal}	0,999923	4,1112	0,68
A_{app}	0,999963	4,4335	0,32
A_{db}	0,995977	2,3955	35,23
A_t	0,995931	2,3905	35,64

Este Estudo de Caso apresentou melhora na métrica de disponibilidade do ambiente, resultando em redução de cerca de 45% no *downtime* anual da aplicação. Porém, segundo os

administradores do sistema, este valor ainda se encontra acima do esperado para a aplicação. Desta forma, é proposto novas modificações no ambiente no próximo Estudo de Caso.

7.4 ESTUDO DE CASO III — APRIMORAMENTO DA DISPONIBILIDADE DO SISTEMA

Este Estudo de Caso irá apresentar um novo cenário para o ambiente estudado, visando aprimorar ainda mais a disponibilidade do sistema. Todos os cenários apresentados até o momento presente foram implementados como solução do ambiente de produção. O ganho e aprimoramento da disponibilidade foi real no sistema estudado, entretanto, deste ponto em diante os modelos são sugestões para melhorar a disponibilidade e não foram realizadas implementações destes modelos no cenário real. Este Estudo de Caso realiza novamente as Macro-atividades 6 (Subseção 4.1.2.1), 7 (Subseção 4.1.2.3) e 8 (Subseção 4.1.2.4), porém com propostas de cenários distintas. Todos os Estudos de Casos se utilizam destas macro-atividades para a sua criação.

Neste momento, são sugeridas mudanças nas camadas de banco de dados. Analisando o Estudo de Caso anterior (Seção 7.3) e na análise de sensibilidade (Subseção 7.2.2), pode-se perceber que a camada de banco de dados apresenta a menor disponibilidade de todo o sistema. Este fato reflete negativamente na disponibilidade total do sistema. Desta forma, é proposto para este Estudo de Caso a duplicação dos componentes da camada de banco de dados. A estrutura do ambiente utilizada neste estudo é similar a apresentada no estudo anterior, exposta na Figura 36. A única modificação entre os modelos é a inclusão de um componente de banco de dados, tornando a totalidade de dois componentes operando em ativo/ativo. Os demais componentes permanecem os mesmos. Para não apresentar uma ilustração idêntica ao estudo anterior, é exposto na Figura 39 apenas o modelo da camada de banco de dados que sofreu alterações.

A adição de uma VM de banco de dados está inclusa no mesmo preceito sobre recursos utilizados anteriormente. Assim, mesmo possuindo uma capacidade maior das suas configurações, um servidor físico suporta apenas uma VM da camada de banco de dados. Sabendo disto, foi necessário realizar a configuração no modelo SPN permitindo apenas uma VM por máquina física. Sempre que houver um problema em um servidor que hospeda o banco de dados, a VM precisa ser automaticamente desligada e não consegue ser iniciada em outro servidor. Esta VM que foi forçada a ser desligada, só pode estar ativa novamente após a restauração do servidor em que estava hospedada. Este controle é realizado através da expressão de guarda exposta na

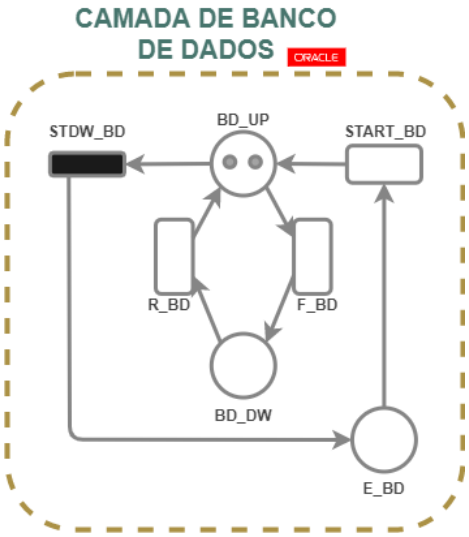


Figura 39 – Estudo de Caso III — Camada de banco de dados.

Tabela 23. Na referida expressão de guarda, o termo $((\#SRV3\_UP) < (\#DB\_UP))$ dispara a transição imediata sempre que o número de *tokens* em DB_UP, que sinaliza a quantidade de VMs operacionais, for maior que o número de *tokens* no lugar SRV3_UP, que representa a quantidade de servidores funcionais. Os demais termos da expressão já foram explicados na Seção 6.2.

Tabela 23 – Estudo de Caso III - Expressões de guarda da camada de banco da dados.

Transição	**Expressão de guarda**
$STDW\_DB$	$(\#SAN\_UP = 0)$ or $(\#STG\_UP = 0)$ or $((\#SRV3\_UP) < (\#DB\_UP))$

As expressões utilizadas para efetuar os cálculos da disponibilidade do modelo são as mesmas apresentadas no Estudo de Caso anterior (Tabela 21). Após a análise estacionária na ferramenta Mercury, obteve-se a disponibilidade mostrada na Tabela 24. Os valores das camadas não expostas na referida tabela não sofreram alterações em relação ao Estudo de Caso anterior (Tabela 22). O valor da disponibilidade encontrado referente ao terceiro Estudo de Caso é $\boldsymbol{A_t = 0,999890}$, possuindo um *downtime* anual de $\boldsymbol{0,96\,h}$. Este valor alcançado como métrica de disponibilidade é considerado aceitável para um sistema de alta disponibilidade operando em nuvem.

Tabela 24 – Estudo de Caso III - Valores da disponibilidade

Métrica	**Disponibilidade**	**#9's**	**dt anual(h)**
A_{db}	0,999942	4,2411	0,50
A_t	0,999890	3,9590	0,96

7.5 ANÁLISE DA EVOLUÇÃO DA DISPONIBILIDADE

Todos os estudos de caso realizados até o momento focam suas sugestões de cenários em alterações de componentes internos ao CD, como: servidores físicos e quantidade de máquinas virtuais. Com essa abordagem, se obteve uma melhoria na disponibilidade classificada como satisfatória para os administradores do sistema.

Os próximos cenários sugerem estruturas de redundância do CD. Porém, antes de dar continuidade, uma breve análise da evolução das disponibilidades até aqui encontradas é realizada. Uma análise dos resultados obtidos nos Estudos de Casos I, II e III está apresentado na Tabela 25.

Tabela 25 – Comparação dos resultados dos Estudos de Caso I, II e III.

(a)

Camadas	#9's			dt anual(h)		
	Caso 01	Caso 02	Caso 03	Caso 01	Caso 02	Caso 03
Conectividade	5,8908	5,8908	5,8908	0,01	0,01	0,01
Armazenamento	4,5123	4,5123	4,5123	0,27	0,27	0,27
Orquestração	2,8233	3,3704	3,3704	13,16	3,73	3,73
Balanceador	2,9369	4,1112	4,1112	10,13	0,68	0,68
Aplicação	2,9323	4,4335	4,4335	10,24	0,32	0,32
Banco de dados	2,2866	2,3955	4,2411	45,28	35,24	0,50
Sistema	2,1366	2,3905	3,9590	63,96	35,64	0,96

(b)

Camadas	Disponibilidade		
	Caso 01	Caso 02	Caso 03
Conectividade	0,999999	0,999999	0,999999
Armazenamento	0,999969	0,999969	0,999969
Orquestração	0,998498	0,999574	0,999574
Balanceador	0,998843	0,999923	0,999923
Aplicação	0,998831	0,999963	0,999963
Banco de dados	0,994831	0,995977	0,999943
Total	0,992698	0,995931	0,999890

Esta análise relaciona os valores de disponibilidade, quantidades de noves (#9's) e tempo de indisponibilidade anual (dt anual) dos Estudos de Casos já apresentados. Os valores da evolução do tempo de indisponibilidade estão expostos na Tabela 25a e da disponibilidade na Tabela 25b, segmentados de acordo com seus respectivos Estudos de Casos, facilitando a identificação individual.

A representação gráfica para a evolução da disponibilidade da aplicação entre os Estudos de Casos está exposta na Figura 40. Nela, pode-se identificar a evolução da disponibilidade nos estudos de caso, principalmente nas camadas que sofreram interferências diretas, como as máquinas virtuais e a camada de banco de dados.

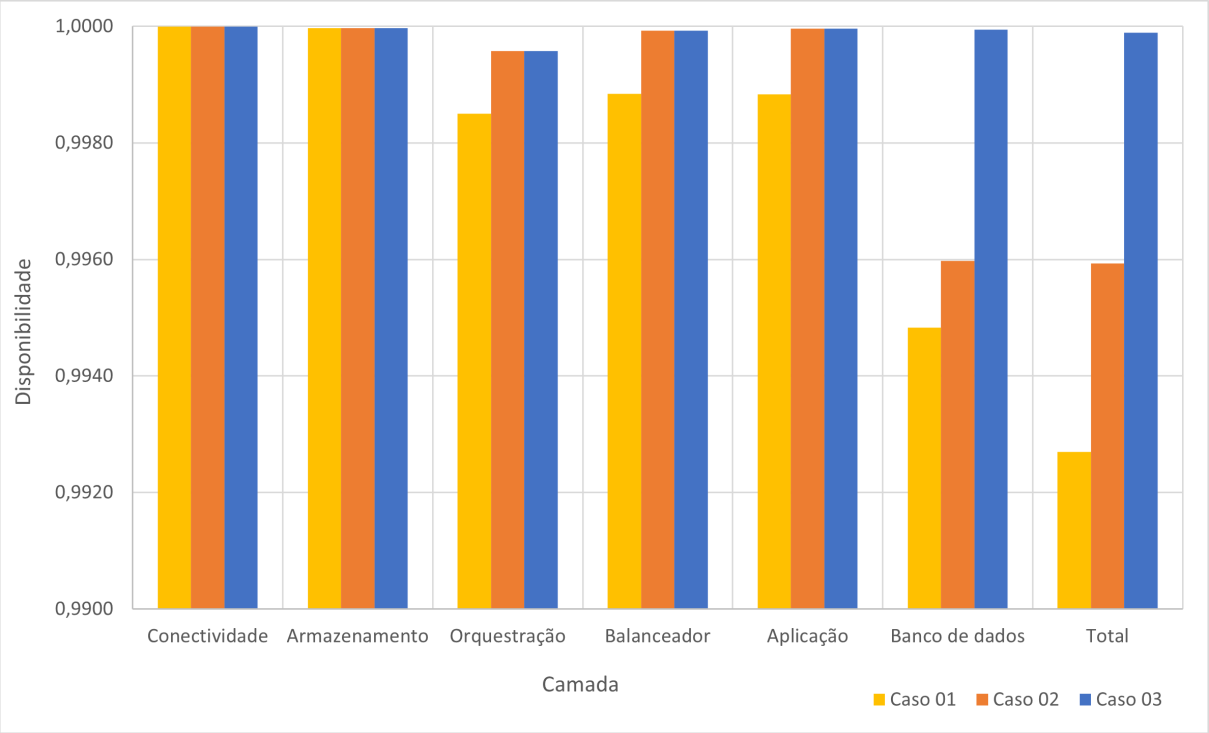


Figura 40 – Comparação dos resultados dos Estudos de Caso I, II e III.

A Figura 41 apresenta uma análise de variação percentual da disponibilidade, possibilitando a identificação do percentual de melhora na métrica da disponibilidade alcançado através dos Estudos de Casos.

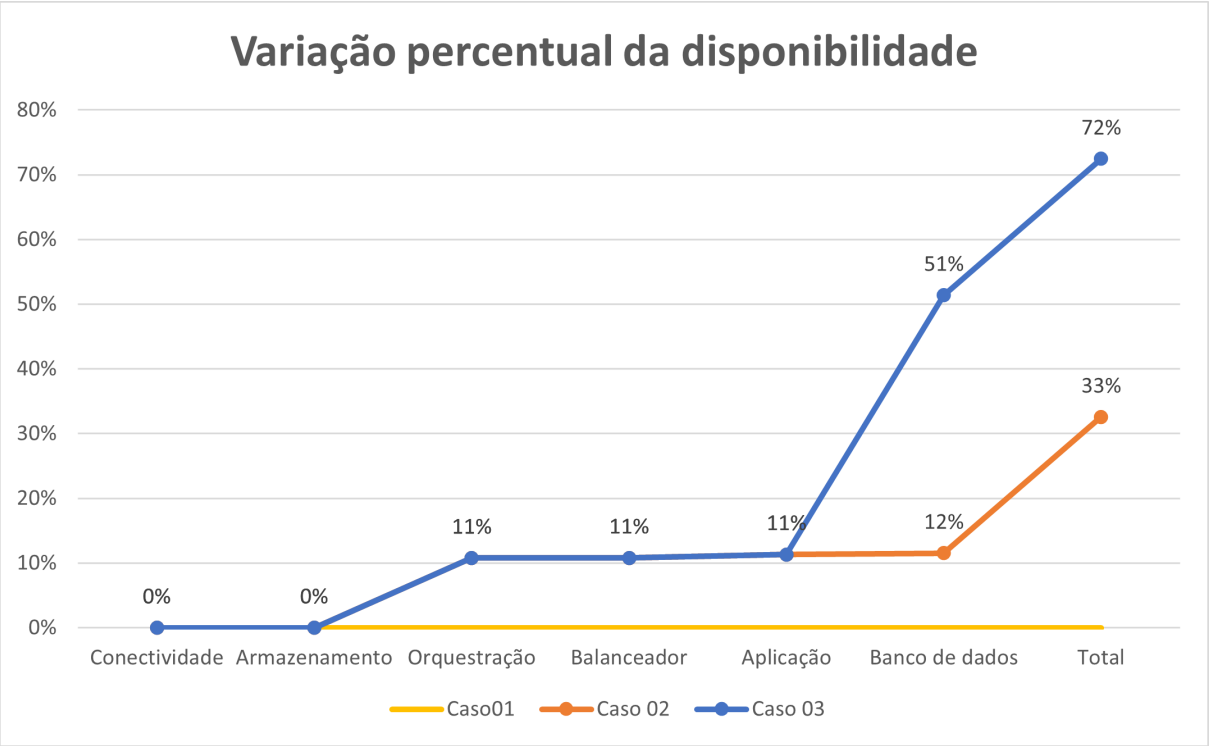


Figura 41 – Variação percentual da disponibilidade dos Estudos de Caso I, II e III.

7.6 ESTUDO DE CASO IV — AVALIAÇÃO DA DISPONIBILIDADE DO CENTRO DE DADOS

Até o momento, todos os cenários propostos analisaram a estrutura interna ao CD. Neste estudo, foi analisanda a disponibilidade do sistema incluindo a estrutura do CD (DCS). Visando atingir este objetivo, um modelo foi proposto para representar o Centro de Dados como unidade única. Este modelo contempla as estruturas de DCS, computacional e lógica. Estas estruturas já foram apresentadas neste estudo nas Seções 5.1, 5.2 e 5.3, respectivamente.

Incluir a estrutura do DCS no modelo SPN apresentado no Estudo de Caso III, traria um custo computacional elevado. A alternativa utilizada na modelagem em SPN para implementar a estrutura de DCS sem alto custo computacional, foi adotar a técnica de simplificação. O modelo apresentado neste estudo é resultado de uma simplificação de segundo nível, pois é utilizada esta técnica duas vezes. O resultado deste processo é um modelo em SPN conciso que representa a Estrutura do Centro de Dados (DCS) e a Estrutura do Sistema estudado (SS), contemplando interferências externas como desastres naturais, por exemplo.

O primeiro nível da simplificação é utilizado para sintetizar o modelo construído no Estudo de Caso apresentado na Seção 7.4. Foi adicionado a esta sintetização o modelo representando o DCS, propondo assim um novo modelo que simbolize o CD. A segunda simplificação é aplicada neste modelo recém-construído, visando a representação do CD contemplando o CCF.

Assim como nas estruturas computacional e lógica, os valores do DCS foram obtidos através do monitoramento que perdurou aproximadamente um ano. A configuração do monitoramento foi semelhante ao apresentado anteriormente neste estudo. Foram identificados e registrados os tempos de falha e reparo de cada componente. Estes tempos estão apresentados no Apêndice A.4. Em posse dos TTFs e TTRs da referida estrutura, se calculou os tempos de MTTF e MTTR do DCS utilizando as Equações 2.9 e 2.10, respectivamente.

Na descoberta dos valores da SS, foram aplicadas técnicas já expostas no estudo apresentado na Seção 6.3. Os tempos referentes ao CCF foram inferidos por análise intuitiva. No mesmo período de análise do DCS, ocorreu apenas um incidente classificável como CCF, que perdurou cerca de 24h. Os tempos de MTTF e MTTR utilizados neste Estudo de Caso estão expostos na Tabela 26, porém, eles podem variar conforme o cenário desejado para o estudo.

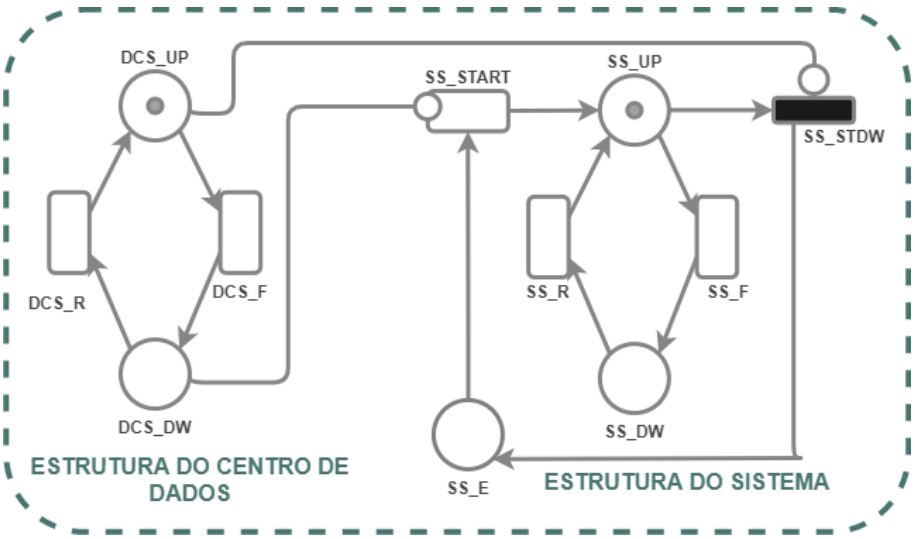
Tabela 26 – Tempos de MTTF e MTTR utilizados no modelo em SPN - Estudo de Caso IV.

Componente	MTTF(h)	MTTR(h)
Estrutura do Centro de Dados (DCS)	2365,25	5,50
Estrutura do Sistema (SS)	183227,80	20,14
Causa Comum de Falha (CCF)	8760,00	24,00

7.6.1 Simplificação do modelo

Nesta fase, foi elaborado um modelo de disponibilidade em SPN que representa o DCS e o SS. O objetivo deste modelo é descobrir a disponibilidade total do sistema, considerando toda a estrutura do CD. Este modelo possui o primeiro nível de simplificação deste estudo, utilizada para sintetizar a estrutura do sistema (estruturas computacional e lógica). Por ser uma sintetização, o valor da disponibilidade do modelo gerado na sintetização deve ser o mesmo do modelo completo, apresentado na Seção 7.4. Os valores do MTTR se refere ao somatório dos MTTRs dos componentes, já o valor do MTTF é obtido através da Equação 2.12.

A representação do modelo de disponibilidade para as estruturas de centro de dados e sistema foi exposta na Figura 29, localizada na Página 97, e seu funcionamento detalhado foi mostrado na Seção 6.3. Para melhor entendimento, fica reapresentada a mesma Figura.



A estrutura de centro de dados está representada no modelo por DCS. Já a simplificação pode ser identificada no grupo definido como estrutura do sistema, que consta no modelo por SS. Conforme já explicado no Capítulo 5, existe uma relação entre os sistemas, pois, caso o

DCS apresente falhas, todo a SS também está no estado de falha. Consequentemente, a SS só pode retornar ao estado operacional caso o DCS também estiver operacional. Foi utilizado o mesmo conceito dos modelos apresentados neste trabalho e criado um lugar especial para identificar quando a SS está inativa por problemas externos, falha no DCS. Da mesma forma foram criadas ações para iniciar e desligar a estrutura SS.

No modelo deste Estudo de Caso, a disponibilidade do DCS é calculada pela probabilidade do referido grupo estar no estado operacional. Já a disponibilidade total pode ser calculada através da probabilidade do grupo SS estar operacional, pois como existe a correlação entre os grupos, indicando que, caso a SS esteja operacional consequentemente o DCS também estará. As expressões para o cálculo da disponibilidade estão expostas na Tabela 27.

Tabela 27 – Estudo de Caso IV - Expressões para o cálculo da disponibilidade.

Métrica	**Expressão**
A_{dcs}	$P\{(\#DCS\_UP > 0)\}$
A_{ss}	$P\{(\#SS\_UP > 0)\}$

Nos modelos anteriores, foram utilizadas expressões de guarda que serviram para inibir determinados disparos das transições. Neste modelo foram utilizados arcos inibidores e expressões de guarda. Esta estratégia não permiti que a SS esteja operacional caso o DCS esteja em falha. O arco inibidor, conectado do lugar DCS_UP à transição SS_STDW, não permite que a transição seja disparada enquanto o DCS esteja no estado operacional. O outro arco inibidor, conectado do lugar DCS_DW à transição SS_START, não permite que a transição SS_START seja disparada caso o DCS esteja em estado de falha. Porém, mesmo com estes arcos inibidores inibindo algumas ações, ainda foi necessário a utilização da expressão de guarda na transição SS_R, pois a SS pode estar no estado de falha, um *token* em SS_DW e tentar iniciar no momento que o DCS estiver em falha. Assim, a expressão $\#DCS\_UP > 0$ foi adicionada em SS_R. Após realizar uma análise estacionária na ferramenta Mercury, obtivemos a disponibilidade mostrada na Tabela 28.

Tabela 28 – Estudo de Caso IV - Valores da disponibilidade para simplificação.

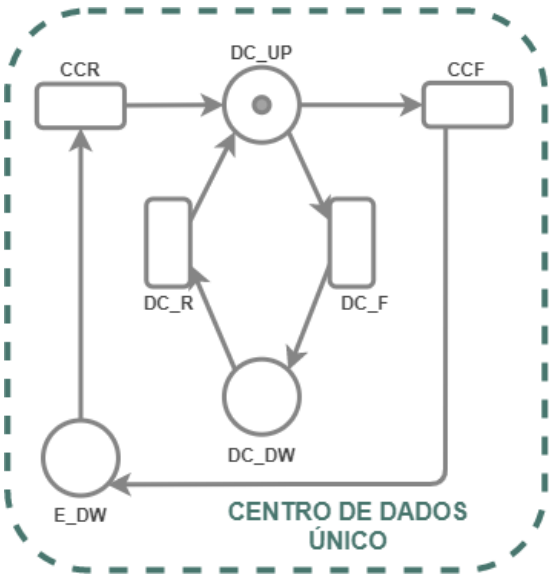
Métrica	**Disponibilidade**	**#9's**	**dt anual(h)**
A_{dcs}	0,997680	2,6345	20,32
A_t	0,997360	2,5783	23,13

A métrica de disponibilidade encontrada no modelo foi de $A_t = 0,997360$, possuindo um *downtime* anual de $23,13\,h$. Este valor é referente à disponibilidade do sistema contemplando a estrutura do CD. Após esta análise, pode-se aprimorar ainda mais o modelo adicionando informações sobre CCF.

7.6.2 Modelo de disponibilidade do centro de dados único

Neste ponto do Estudo de Caso, foi realizado o segundo nível de simplificação visando criar um modelo de disponibilidade que represente o CD e o sistema estudado. O modelo concebido nesta fase do estudo contempla a CCF e será utilizado na formação da estrutura do modelo apresentado no próximo Estudo de Caso. Para adicionar o CCF no modelo de uma maneira mais eficiente e com menos processamento nas análises, foi realizada uma nova sintetização do modelo apresentado na Subseção 7.6.1 e implementado informações sobre o CCF.

Utilizou-se o mesmo procedimento adotado anteriormente para realizar a sintetização, aplicando as técnicas já expostas neste estudo da Seção 6.3 para encontrar o MTTF do CD. A sintetização realizada neste momento cria um modelo simples que representa o CD, possibilita a implementação de CCF e redundâncias. A Figura 30 mostra o modelo, localizada na Página 98. A referia Figura está reapresentada nesta parte do trabalho visando um melhor entendimento. Os tempos de MTTF e MTTR utilizados no modelo estão expostos na Tabela 29.



A expressão utilizada para encontrar a disponibilidade na ferramenta Mercury foi $\boldsymbol{A_{(t)} = P\{(\#DC\_UP > 0)\}}$. Após análise estacionária, foi calculado a disponibilidade do ambi-

Tabela 29 – Tempos médios de utilizados no modelo em SPN - Estudo de Caso IV.

Componente	**MTTF(h)**	**MTTR(h)**
Centro de Dados (DC)	9683,90	25,64
Causa Comum de Falha (CCF)	8760,00	24,00

ente e o resultado está exposto na Tabela 30.

Tabela 30 – Estudo de Caso IV — Valores da disponibilidade para centro de dados único.

Métrica	**Disponibilidade**	**#9's**	**dt anual(h)**
A_t	0,994604	2,2679	47,26

O valor da métrica de disponibilidade para este estudo é $\boldsymbol{A_t = 0,994604}$ com um tempo de indisponibilidade anual em horas de $\boldsymbol{47,26\,h}$. Pode-se perceber que o valor da disponibilidade e do tempo de indisponibilidade pioraram em comparação com o modelo anterior, pelo fato de ter sido adicionado os valores do CCF. O valor de disponibilidade encontrado neste estudo está mais próximo da realidade do ambiente estudado. O valor desta disponibilidade ainda é baixa em comparação com outros sistemas hospedados na nuvem. Visando aprimorar essa métrica, foi realizado outro Estudo de Caso implementando a replicação do CD.

7.7 ESTUDO DE CASO V — AVALIAÇÃO DA DISPONIBILIDADE DO CENTRO DE DADOS REDUNDANTE

A estrutura de um CD é diretamente proporcional à disponibilidade de serviços hospedados. O CD estudado não se encaixa perfeitamente nas descrições de um CD *Tier* II, pois não possui duas empresas de telecomunicações nem dois geradores para fornecimento do sistema energético. No entanto, os demais recursos e sua disponibilidade se aproximam com as características dos CDs Tier II. Existem alternativas amplamente estudadas que visam aprimorar a disponibilidade de um CD, porém as duas que se destacam são: melhorar a classificação do CD, melhorando as redundâncias internas ou implantar a redundância do CD. Cada uma das configurações possui pontos positivos e negativos. Porém, a redundância de CD possibilita, além do aprimoramento da métrica estudada, a utilização de alta disponibilidade nos serviços hospedados, balanceamento de carga entre CDs, melhoramento do armazenamento e políticas de *backup* e, principalmente, possibilita a recuperação de desastres.

O objetivo deste Estudo de Caso é analisar possíveis cenários de redundância do CD estudado, identificando a melhor estratégia de redundância. Para atingir este objetivo, foram propostos cenários com diferentes configurações de redundância do CD, permitindo a identificação de quais CDs estão ativos e recebendo as requisições dos clientes. Como ferramenta de análise, foram concebidos modelos de disponibilidade em SPN que facilitam a tomada de decisão na escolha para o melhor cenário.

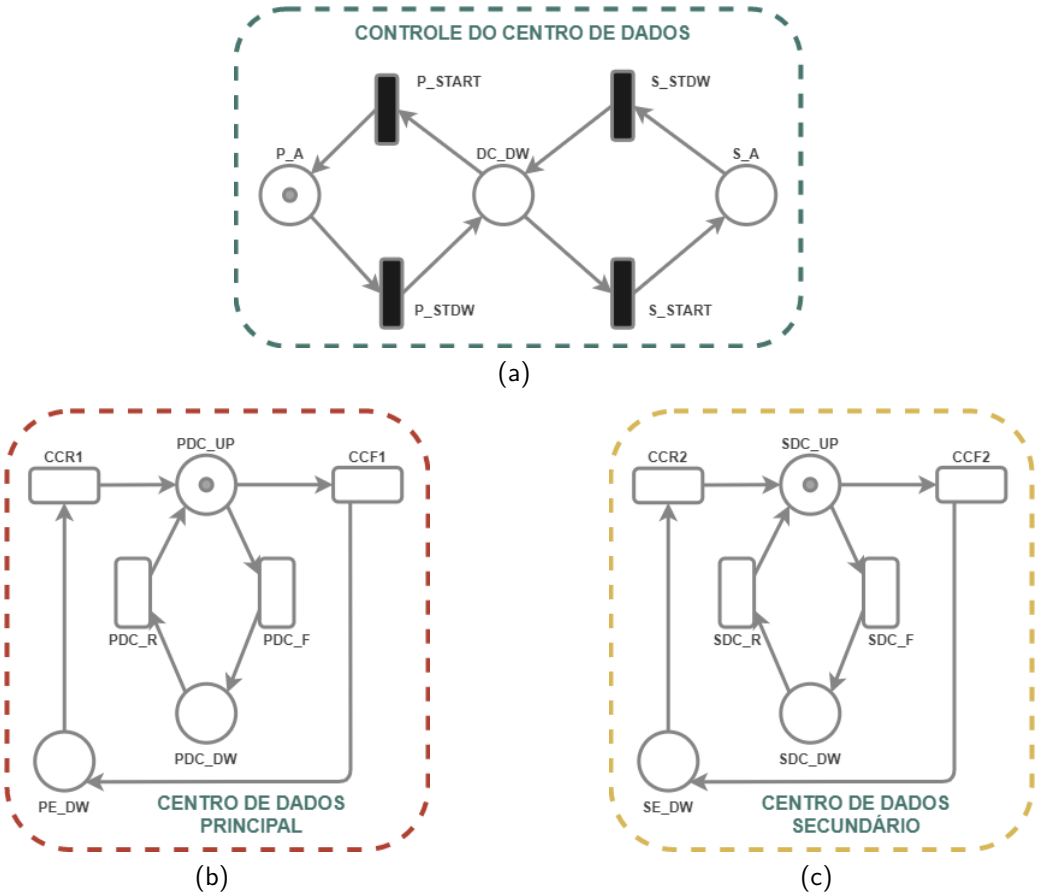
Este trabalho, até o presente Estudo de Caso, foi baseado no monitoramento dos componentes do ambiente. Porém, neste Estudo de caso, esta utilização não foi possível por não existirem dois CDs para serem monitorados. Desta forma, a estratégia utilizada foi a replicação dos dados referentes ao CD existente, para o CD secundário. A partir deste momento do estudo, o CD existente foi classificado como primário e os fictícios como secundários. Os CDs operam independentes. Desta forma, caso um CD esteja indisponível, não interfere na disponibilidade dos demais CDs. As questões relacionadas a transferência de dados, *throughput* da rede e estudos geográficos não foram considerados no estudo. Supõe-se que os dados estão sendo constantemente replicados e não há perda na replicação.

Foram realizados estudos comparativos de disponibilidade em diferentes cenários de redundância do CD. O **Cenário 01** possui *dois CDs operando em ativo/passivo*. Esta configuração possui apenas um CD ativo por vez, tendo suas requisições externas direcionadas para apenas um CD. Mesmo que o outro CD esteja operacional, não poderá estar ativo. Este controle é realizado pelo modelo de controle de CD.

O **Cenário 02** possui *dois CDs operando em ativo/ativo*. Esta configuração permite a existência de dois CD ativos e recebendo as requisições externas simultaneamente, havendo um balanceamento de carga entre os CDs. Caso um dos CDs apresente falha, todas as requisições são redirecionadas para o CD que permanece funcional.

O **Cenário 03** possui *Três CDs operando em ativo/passivo*. Esta configuração é uma junção das duas apresentadas anteriormente, pois permite o balanceamento de carga entre os CDs e ainda uma redundância, caso um CD apresente falha. A redundância respeita o máximo de dois CDs ativos por vez, tendo um CD em modo de espera. Caso um CD ativo apresente falha, as requisições externas passam para o terceiro CD, continuando com dois CDs ativos.

Os modelos aqui apresentados sofrerão alterações conforme o cenário estudado. Para a estrutura básica, Cenário 01, a apresentação do modelo de disponibilidade foi exibido na Figura 31, localizado na Página 100, porém, é reapresentado para melhor entendimento.



O modelo proposto possui três estruturas básicas, o Controle do Centro de Dados (CCC), o Centro de Dados Principal (PDC) e o Centro de Dados Secundário (SDC). Delas, apenas a estrutura referente ao grupo do Controle do Centro de Dados não foi apresentada. As demais foram o resultado do Estudo de Caso IV. A estrutura do Controle do Centro de Dados possui a função de identificar o estado dos CDs e redirecionar as solicitações externas para o CD ativo. Os lugares P_A e S_A identificam o CD ativo. Os *tokens* no lugar P_A identificam que o CD principal está ativo. Por outro lado, o *token* no lugar S_A, sinaliza que o CD secundário está ativo. O lugar DC_DW é um estado de transição e indica que um CD que estava ativo apresentou falha.

O PDC está com estado de ativo se houver *tokens* no lugar PDC_UP. Por outro lado, o SDC recebe o estado de ativo se estiver operacional e o PDC falhar. Um processo semelhante ocorre para retornar o PDC como ativo. É necessário que o SDC esteja no estado de falha e o PDC esteja operacional. O número de tokens no lugar PDC_UP representa o número de CDs ativos. Neste Estudo de Caso, a quantidade máxima de tokens neste lugar são dois tokens (Cenário 03). No entanto, pode-se quantificar o número de CDs ativos que quisermos, diferente do SDC que tem sempre um token, pois representa um CD secundário (CD de *backup*).

O modelo recém-apresentado sofre pequenas alterações conforme o cenário deste Estudo de Caso, porém, a expressão de cálculo da disponibilidade permanece a mesma para todos os cenários, Tabela 31 , assim como as expressões de guarda que já foram apresentadas na Tabela 9, localizada na Página 100.

Tabela 31 – Estudo de Caso V - Expressões para o cálculo da disponibilidade.

Métrica	**Expressão**
A_t	$P\{(\#P\_A > 0)$ or $(\#S\_A > 0)\}$

No Segundo cenário se utilizou dois CDs operando em ativo/ativo. Desta forma, o grupo controle do centro de dados sofreu alterações, possuindo um *token* em P_A e outro em S_A, apresentado na Figura 42. Os demais grupos permaneceram sem alterações.

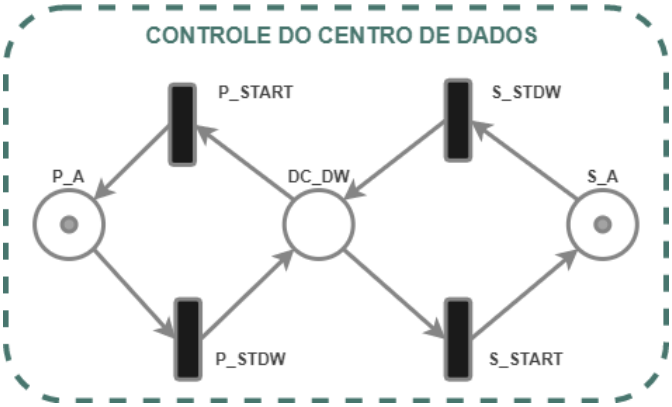


Figura 42 – Estudo de Caso V — Cenário 02 — Controlador do centro de dados.

O terceiro cenário utiliza três CDs, sendo dois operando de forma ativo e um de forma passiva. Deste modo, os modelos representando os grupos de controle do centro de dados e do CD principal sofrem alterações, sendo apresentados conforme a Figura 43.

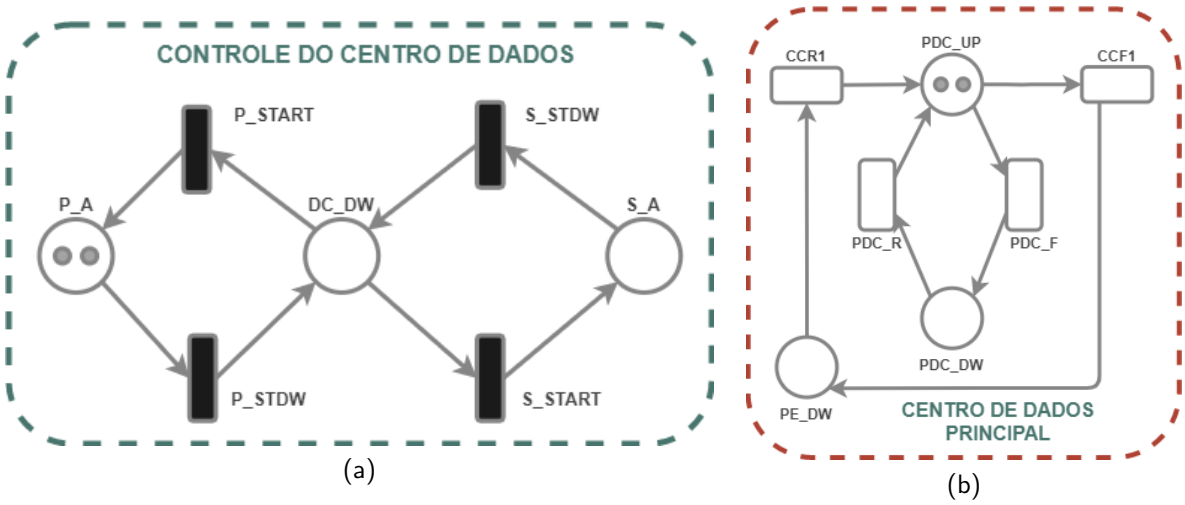


Figura 43 – Estudo de Caso V — Cenário 03 — Controlador do CD e CD principal.

Para este cenário o modelo do controlador recebe dois *tokens* no lugar P_A, representando dois CDs primários e ativos. O modelo representando o PDC também recebe dois *tokens* no lugar PDC_UP, sinalizando que existem dois CDs operando em ativo/ativo, e realizando o balanceamento de carga.

Após a análise estacionária em cada cenário, se alcançou os valores de disponibilidade do ambiente. Os resultados estão na Tabela 32.

Tabela 32 – Estudo de Caso V — Valores da disponibilidade

Cenários	Tipo de redundância	Disponibilidade	#9's	dt anual(h)
1	2 CDs (1 Ativo / 1 Passivo)	0,999971	4,5419	0,25
2	2 CDs (1 Ativo / 1 Ativo)	0,999978	4,6645	0,19
3	3 CDs (2 Ativo / 1 Passivo)	0,999999	6,9360	0,0001

Pode-se perceber que a disponibilidade utilizando redundância de CD consegue atingir índices altos. No entanto, a diferença entre os cenários que se utilizam de dois CDs, atuando de modo ativo/ativo para ativo/passivo é pequena. Ao usar três CDs, dois ativos e um em espera, o valor de disponibilidade aumenta drasticamente.

7.8 ANÁLISE DA EVOLUÇÃO DA DISPONIBILIDADE DOS ESTUDOS DE CASOS

Esta seção apresenta uma análise dos resultados obtidos nos Estudos de Caso. Como explicado na primeira Análise Evolutiva na Seção 7.5, os Estudos de Casos deste trabalho inicialmente propuseram cenários visando melhorias internas ao CD (Estudos de Casos I, II e III). Ao obter um valor considerado satisfatório para os responsáveis do sistema estudado, os Estudos de Casos foram voltados à estrutura do CD e sua replicação (Estudos de Casos IV e V).

Uma análise comparativa de variação percentual da disponibilidade para os Estudos de Casos IV e IV está apresentada na Figura 44. Nela, pode-se identificar o percentual de evolução da métrica da disponibilidade ao realizar a redundância do CD.

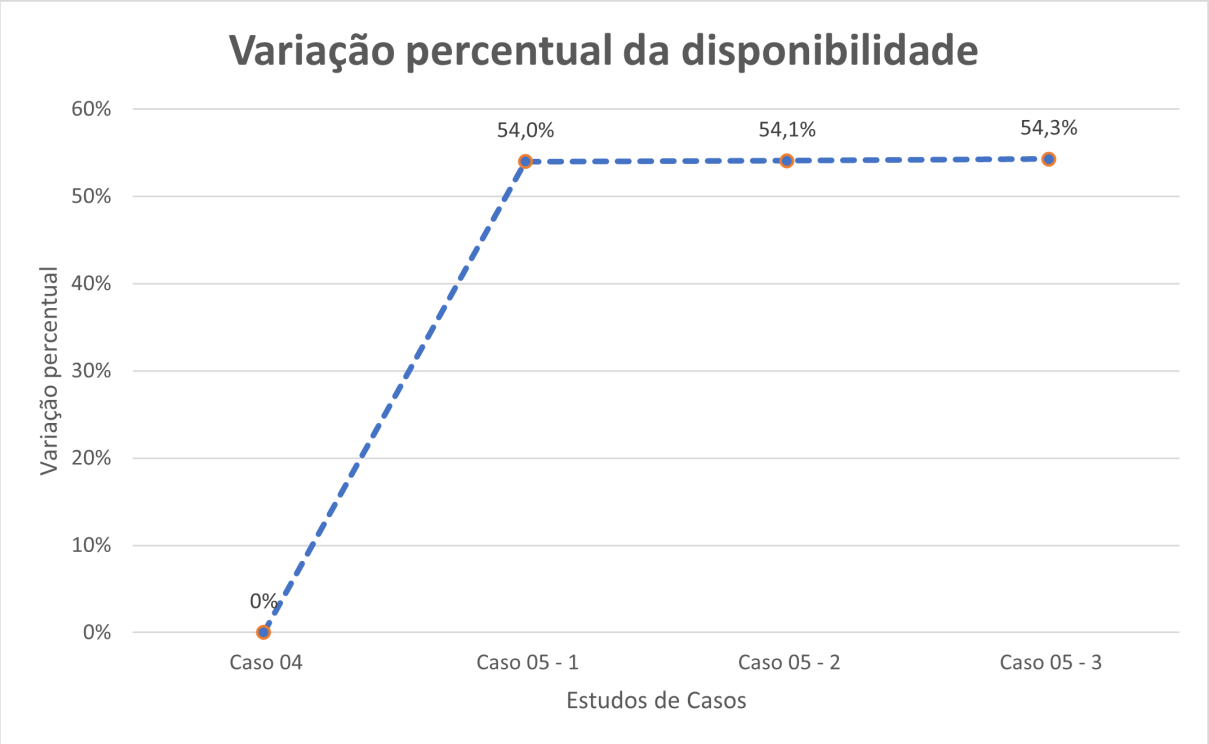


Figura 44 – Variação percentual da disponibilidade dos Estudos de Caso IV e V.

A comparação da métrica da disponibilidade, quantidade de noves e *downtime* anual de todos os Estudos de Casos está exposta de forma numérica na Tabela 33 e de forma visual nos gráficos apresentados nas Figuras 45, 46 e 47.

Tabela 33 – Análise evolutivo da disponibilidade nos Estudos de Casos.

Estudo de Caso	Disponibilidade	#9's	dt anual(h)
1	0,99279	2,1365	63,96
2	0,99593	2,3905	35,64
3	0,99989	3,959	0,96
4	0,99460	2,2679	47,26
5.1	0,99997	4,5419	0,25
5.2	0,99997	4,6645	0,19
5.3	0,99999	6,936	0,0001

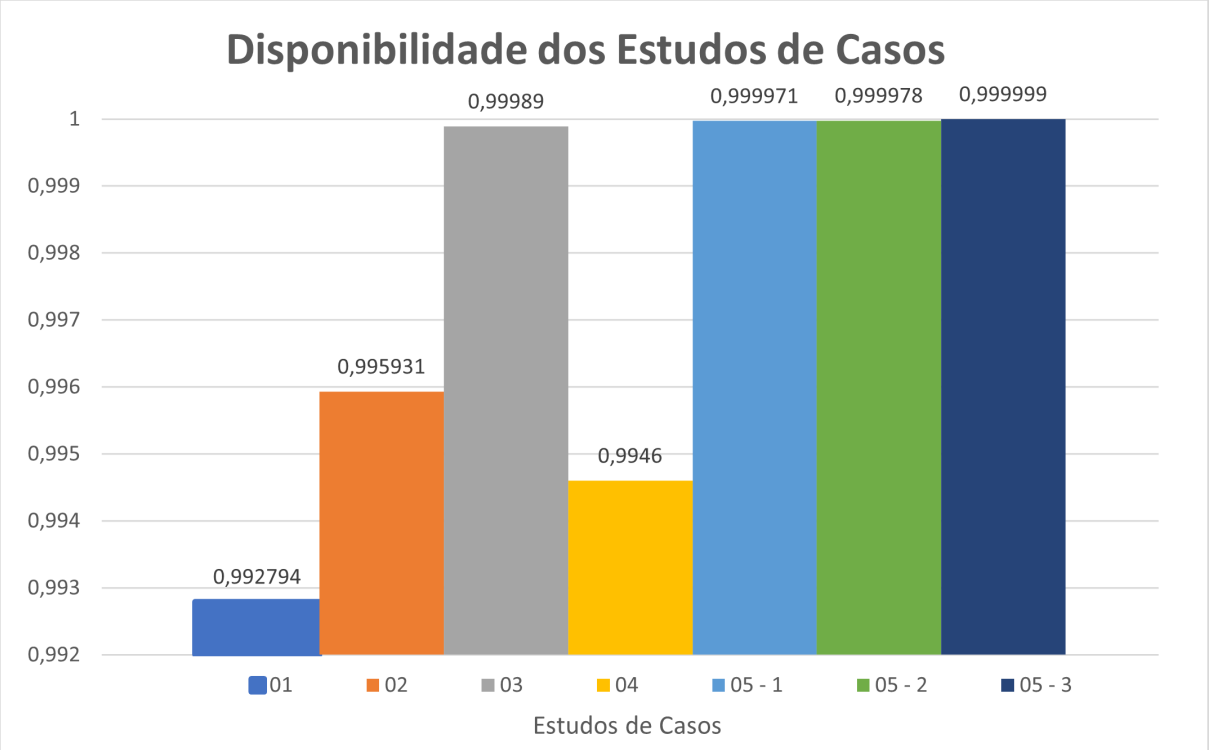


Figura 45 – Gráfico evolutivo da disponibilidade obtidas nos Estudos de Casos.

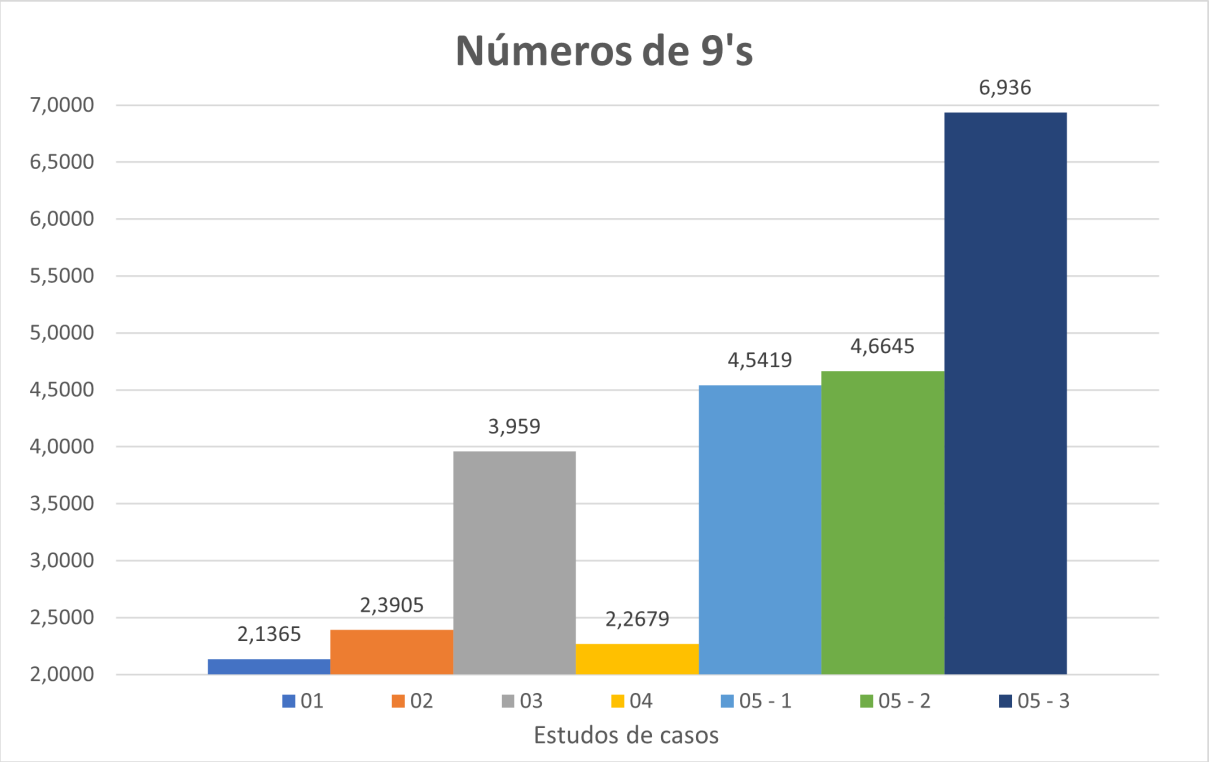


Figura 46 – Gráfico evolutivo do número de noves obtidos nos Estudos de Casos.

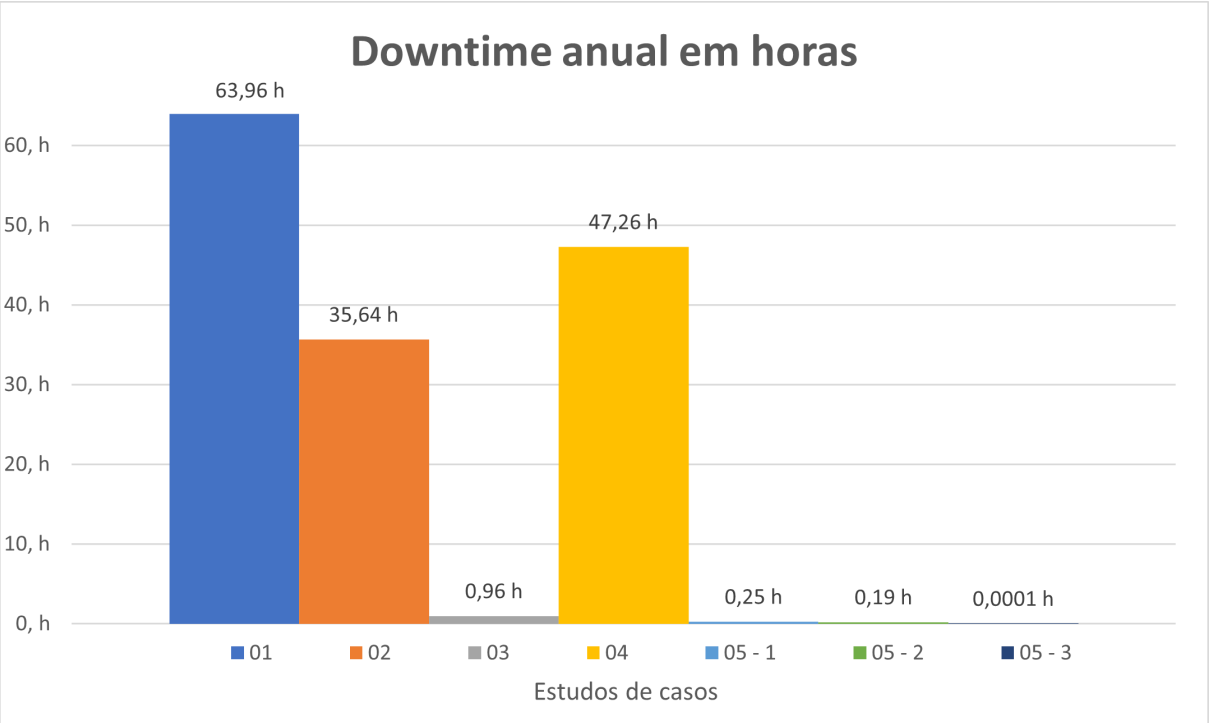


Figura 47 – Gráfico evolutivo do *downtime* em horas obtidos nos Estudos de Casos.

Pode-se perceber visualmente a evolução da disponibilidade obtidas através dos modelos propostos, juntamente com a diminuição drástica do tempo de indisponibilidade da aplicação estudada. A melhoria da métrica disponibilidade em forma percentual, também pode ser obser-

vada na Figura 48, que se refere a uma comparação da variação percentual da disponibilidade de todos os Estudos de Casos.

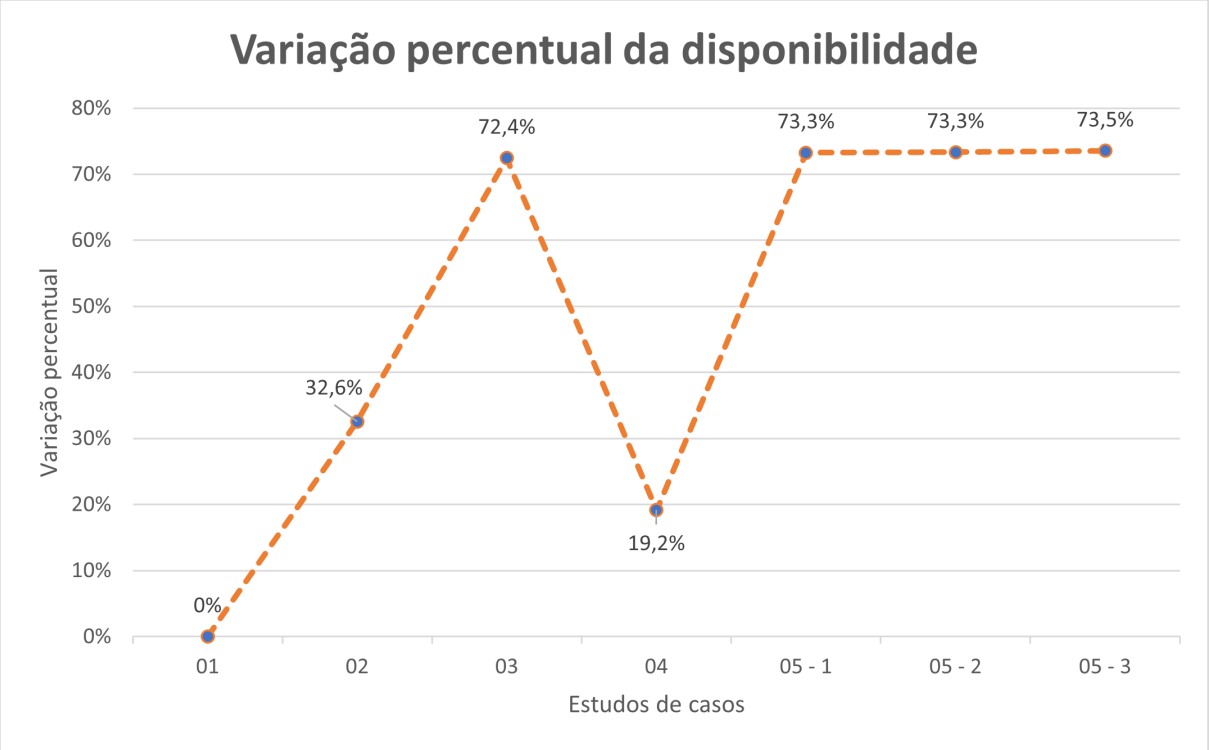


Figura 48 – Variação percentual da disponibilidade dos Estudos de Casos.

8 CONCLUSÃO E TRABALHOS FUTUROS

A computação em nuvem se tornou amplamente utilizada por instituições das mais diversas estruturas, que visam ofertar aos seus clientes um serviço de qualidade e que esteja sempre disponível. Porém, os desafios para utilização de uma infraestrutura em nuvem são tão extensos quanto suas vantagens. Os responsáveis por administrar os serviços em nuvem precisam se preocupar em melhorar o aproveitamento dos recursos físicos e energéticos, armazenamento e replicação de dados, alta disponibilidade, recuperação em casos de desastres, dentre outros pontos essenciais para o correto funcionamento dos serviços hospedados.

O presente trabalho, analisa um sistema responsável pela gerência acadêmica de alunos, que se utiliza de um CD utilizando mecanismos de nuvem privada. O objetivo principal deste estudo foi aprimorar a disponibilidade do sistema, propondo ajustes nas infraestruturas lógicas e físicas do ambiente. Estes ajustes passam de simples restruturação na quantidade de máquinas virtuais que ofertam um determinado serviço, à identificação da melhor configuração para a replicação de CD visando obter um alto índice da sua disponibilidade.

Recorrendo à metodologia aqui apresentada, este estudo pode ser replicado para outros ambientes que possuam características semelhantes. A análise prévia da infraestrutura utilizada, permitiu melhores ajustes na ferramenta de monitoramento, consequentemente, apresentando dados mais condizentes com a realidade. Alguns dos estudos de casos aqui apresentados, foram replicados no ambiente real de produção e seus resultados foram o aumento no índice da disponibilidade do sistema.

Este trabalho conduz cinco estudos de casos que se utilizaram de modelagens para planejar a infraestrutura do sistema. Primeiro, foi construído um modelo que representa o ambiente físico e virtual, contemplando movimentações das VMs entre servidores previamente definidos. Em seguida, foi realizada uma análise para identificar os componentes que possuem a tendência de aprimorar a disponibilidade com menor esforço. Após isto, foram propostos modelos hierárquicos de disponibilidade que visaram aprimorar a disponibilidade do sistema, incluindo redundância de componentes e de CD.

O primeiro estudo apresentou modelos hierárquicos das estruturas computacionais e lógicas utilizadas pelo sistema. Esta estrutura foi denominada como *baseline*. Utilizou-se, também, da criação de intervalo de confiança, através da técnica de *Bootstrapping*, para validar o modelo. O estudo teve como última ação a identificação através da análise de sensibilidade de quais

componentes possuem a tendência mais significativa para alterar a disponibilidade total do ambiente.

O segundo e o terceiro Estudos de Casos propõem ajustes na infraestrutura do sistema estudado, visando melhoria da disponibilidade. Seguindo a análise de sensibilidade, foi percebido que dois componentes possuem alto índice de sensibilidade à disponibilidade: o *banco de dados* e o *servidor físico*. Desta forma, foram aplicadas redundâncias nestas camadas visando uma melhora na disponibilidade total do ambiente. Estes estudos de casos apontaram para uma melhoria significativa na métrica de disponibilidade, tendo o valor da disponibilidade uma relevância de $\mathbf{0,992673}$ com um tempo de indisponibilidade anual em horas de $\mathbf{63,94}\,\boldsymbol{h}$, no Estudo de Caso I, para $\mathbf{0,999890}$ com um tempo de indisponibilidade anual em horas de $\mathbf{0,96}\,\boldsymbol{h}$, no Estudo de Caso III.

O quarto Estudo de Caso aplicou dois níveis de simplificação em SPN, visando alcançar um modelo de baixo processamento computacional e que contemple o CD e CCF. O quinto Estudo de Caso foi a análise de três cenários de redundância CD. Foi criado um modelo de disponibilidade que permitiu a identificação do estado do CD (ativo e passivo).

O cenário recomendado para este estudo não foi o que obteve a melhor disponibilidade, verificada no Cenário 03 do quinto Estudo de Caso. Este estudo se utilizou de dois CDs ativos e um CD passivo. Com essa configuração, se obteve uma disponibilidade de $\mathbf{0,9999998}$, com aproximadamente sete noves e um *downtime* anual de $\mathbf{0,0001}\,\boldsymbol{h}$. Essa disponibilidade é almejada para grandes provedores de serviços em nuvem, o que naturalmente não está no escopo da instituição de ensino que está sendo utilizada neste estudo. Desta forma, as escolhas alcançáveis para a instituição estão entre o Cenário 01 e 02 do Estudo de Caso V. Nestes cenários, os valores das disponibilidades estão muito próximos, indicando que ambos possibilitam uma melhoria da disponibilidade classificada como satisfatória para o sistema estudado e podem ser utilizados no ambiente de produção da Universidade.

Assim, o requisito para escolha do cenário recomendado será o que possibilita melhor balanceamento de carga entre os serviços, por utilizar-se de dois CDs atuando em HA. Então, o Cenário 02 do Estudo de Caso V é a estrutura recomendada por ter atingido a disponibilidade de $\mathbf{0,999978}$, com um *downtime* anual em horas de $\mathbf{0,19}\,\boldsymbol{h}$, possibilitando ainda a utilização de um balanceamento de carga mais efetivo.

A abordagem apresentada neste trabalho pode ser aplicada a serviços hospedados em nuvem privada onde o sistema possua uma infraestrutura semelhante ao do sistema estudado.

Este trabalho foi uma implementação prática em um ambiente que está recebendo requisi-

ções de clientes reais. Alguns dos estudos aqui apresentados foram implementados e os ganhos na disponibilidade foram constatados no ambiente de produção do serviço, representando uma melhoria do serviço ofertado pela aplicação. O Estudo de Caso I, por exemplo, foi inteiramente implementado e a partir dele se pôde identificar pontos de melhorias no sistema. Com a análise de sensibilidade, ficou perceptível que a abordagem que seria implementada inicialmente (aumentar a quantidade de aplicações) seria ineficaz. Se percebeu também que a camada de banco de dados é o principal responsável por diminuir a disponibilidade do ambiente, algo que não estava nítido antes do estudo. O Estudo de Caso II foi implementado de forma distinta, porém, proporcionando a alta disponibilidade da camada de virtualização. Por outro lado, o Estudo de Caso III ainda não foi implementado por questões técnicas em realizar o *cluster* da camada de banco de dados, todavia, já está sendo estudada a solução para o caso.

Outros estudos também apresentados aqui não foram implementados, mas utilizados como tomada de decisão para futuras expansões, como o Estudo de Caso V. Existe um projeto de implementação de site redundante, que inicialmente teria a configuração idêntica ao Cenário 01. Após este trabalho, o projeto foi alterado e será implementada a configuração tomando-se como base o Cenário 02 deste estudo.

8.1 TRABALHOS FUTUROS E LIMITAÇÕES

Embora este trabalho tenha alcançado uma série de avanços relacionados ao estudo de dependabilidade do ambiente analisado, há ainda uma série de possibilidades a serem exploradas.

Primeiramente, pode-se aplicar outras técnicas de análise de replicação de centro de dados, identificando as ferramentas adequadas e os melhores meios de realizar a replicação das informações contidas no centro de dados - do principal para o secundário. Este estudo permite a possibilidade de estender na identificação adequada da localização física para o CD redundante, assim como no estudo de largura de banda necessária para realização eficiente da replicação dos dados entre os CDs. Outra área de pesquisa que pode ser seguida é o estudo do custo financeiro ou operacional quando se utiliza uma solução de redundância de CD, possibilitando identificação dos custos financeiros e operacionais para replicação dos dados, infraestrutura e tecnologia necessárias para utilização de redundância de centro de dados, podendo também realizar comparações entre tipos de configurações totalmente ativo ou ativo e passivo.

REFERÊNCIAS

AC02762480, A. *Dependable computing and fault-tolerant systems*. [S.l.]: Springer-Verlag, 1987.

ALSHAMMARI, M. M.; ALWAN, A. A.; NORDIN, A.; AL-SHAIKHLI, I. F. Disaster recovery in single-cloud and multi-cloud environments: Issues and challenges. In: IEEE. *2017 4th IEEE International Conference on Engineering Technologies and Applied Sciences (ICETAS)*. [S.l.], 2017. p. 1–7.

ANDRADE, E.; NOGUEIRA, B.; MATOS, R.; CALLOU, G.; MACIEL, P. Availability modeling and analysis of a disaster-recovery-as-a-service solution. *Computing*, Springer, v. 99, n. 10, p. 929–954, 2017.

ARMBRUST, M.; FOX, A.; GRIFFITH, R.; JOSEPH, A. D.; KATZ, R.; KONWINSKI, A.; LEE, G.; PATTERSON, D.; RABKIN, A.; STOICA, I. et al. A view of cloud computing. *Communications of the ACM*, ACM New York, NY, USA, v. 53, n. 4, p. 50–58, 2010.

AVIZIENIS, A.; LAPRIE, J.-C.; RANDELL, B. Fundamental concepts of dependability. *Department of Computing Science Technical Report Series*, Newcastle University, 2001.

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*, IEEE, v. 1, n. 1, p. 11–33, 2004.

AVIZIENIS A.; LAPRIE, J.-C. Dependable computing: From concepts to design diversity. *Proceedings of the IEEE*, IEEE, v. 74, p. 629–638, 1986. ISSN 0018-9219,1558-2256. Disponível em: <http://doi.org/10.1109/proc.1986.13527>.

BARROSO, L. A.; CLIDARAS, J.; HÖLZLE, U. The datacenter as a computer: An introduction to the design of warehouse-scale machines. *Synthesis lectures on computer architecture*, Morgan & Claypool Publishers, v. 8, n. 3, p. 1–154, 2013.

BARROSO, L. A.; HÖLZLE, U.; RANGANATHAN, P. The datacenter as a computer: Designing warehouse-scale machines. *Synthesis Lectures on Computer Architecture*, Morgan & Claypool Publishers, v. 13, n. 3, p. i–189, 2018.

BAUER, E. *Design for reliability: information and computer-based systems*. [S.l.]: John Wiley & Sons, 2011.

BAUER, E.; ADAMS, R. *Reliability and availability of cloud computing*. [S.l.]: John Wiley & Sons, 2012.

BAUER, E.; ADAMS, R.; EUSTACE, D. *Beyond redundancy: how geographic redundancy can improve service availability and reliability of computer-based systems*. [S.l.]: John Wiley & Sons, 2011.

BONNEVILLE, D.; PEKELNICKY, R. Structural design in data centers. In: *Data Center Handbook*. John Wiley & Sons, Inc, 2014. p. 245–255. Disponível em: <https://doi.org/10.1002\%2F9781118937563.ch13>.

BRUNEO, D. A stochastic model to investigate data center performance and qos in iaas cloud computing systems. *IEEE Transactions on Parallel and Distributed Systems*, IEEE, v. 25, n. 3, p. 560–569, 2013.

CACUCI, D. G.; IONESCU-BUJOR, M.; NAVON, I. M. *Sensitivity and uncertainty analysis, volume II: applications to large-scale systems*. [S.l.]: CRC press, 2005. v. 2.

CALLOU, G.; ANDRADE, E.; FERREIRA, J. Modeling and analyzing availability, cost and sustainability of it data center systems. In: IEEE. *2019 IEEE International Conference on Systems, Man and Cybernetics (SMC)*. [S.l.], 2019. p. 2127–2132.

ČEPIN, M. Reliability block diagram. In: *Assessment of Power System Reliability*. [S.l.]: Springer, 2011. p. 119–123. ISBN 978-0-85729-687-0.

CLEMENTE, D.; PEREIRA, P.; DANTAS, J.; MACIEL, P. Availability evaluation of system service hosted in private cloud computing through hierarchical modeling process. *The Journal of Supercomputing*, Springer, p. 1–40, 2022.

COLOCATION, A. *Data Center Standards (Tiers I-IV)*. 2021. Disponível em: <https://www.colocationamerica.com/data-center/tier-standards-overview.htm>. Acesso em: Last accessed 31 Dez 2021.

DEVORE, J. L. Probability and statistics for engineering and the sciences. Spinger, 2008.

DHANUJATI, N.; GIRSANG, A. S. Data center-disaster recovery center (dc-drc) for high availability it service. In: IEEE. *2018 International Conference on Information Management and Technology (ICIMTech)*. [S.l.], 2018. p. 55–60.

DILLON, T.; WU, C.; CHANG, E. Cloud computing: issues and challenges. In: IEEE. *2010 24th IEEE international conference on advanced information networking and applications*. [S.l.], 2010. p. 27–33.

DIXON, P. M. Bootstrap resampling. *Encyclopedia of environmetrics*, Wiley Online Library, v. 1, 2006.

EXTREME, N. *Mean Time Between Failures (MTBF) - Extreme Networks*. 2022. Disponível em: <https://www.extremenetworks.com/support/mean-time-between-failures>. Acesso em: Last accessed 27 Out 2021.

FERREIRA, J.; CALLOU, G.; TUTSCH, D.; MACIEL, P. Pldad—an algorihm to reduce data center energy consumption. *Energies*, Multidisciplinary Digital Publishing Institute, v. 11, n. 10, p. 2821, 2018.

FURHT, B.; ESCALANTE, A. et al. *Handbook of cloud computing*. [S.l.]: Springer, 2010. v. 3.

GENG, H. Data centers-strategic planning, design, construction, and operations. In: *Data Center Handbook*. John Wiley & Sons, Inc, 2014. p. 1–14. Disponível em: <https://doi.org/10.1002\%2F9781118937563.ch1>.

GERMAN, R. *Performance analysis of communication systems with non-Markovian stochastic Petri nets*. [S.l.]: John Wiley & Sons, Inc., 2000.

GHANEM, R.; HIGDON, D.; OWHADI, H. *Handbook of uncertainty quantification*. [S.l.]: Springer, 2017. v. 6. ISBN 978-3-319-12384-4.

GONG, C.; LIU, J.; ZHANG, Q.; CHEN, H.; GONG, Z. The characteristics of cloud computing. In: IEEE. *2010 39th International Conference on Parallel Processing Workshops*. [S.l.], 2010. p. 275–279.

GRAY J.; SIEWIOREK, D. High-availability computer systems. *Computer*, IEEE, v. 24, p. 39–48, 1991. ISSN 0018-9162. Disponível em: <http://doi.org/10.1109/2.84898>.

GREENBERG, A.; HAMILTON, J.; MALTZ, D. A.; PATEL, P. The cost of a cloud: Research problems in data center networks. *SIGCOMM Comput. Commun. Rev.*, Association for Computing Machinery, New York, NY, USA, v. 39, n. 1, p. 68–73, dez. 2009. ISSN 0146-4833. Disponível em: <https://doi.org/10.1145/1496091.1496103>.

GUO, H.; YANG, X. A simple reliability block diagram method for safety integrity verification. *Reliability Engineering & System Safety*, Elsevier, v. 92, n. 9, p. 1267-1273, 2007.

GUPTA, A.; CHRISTIE, R.; MANJULA, P. Scalability in internet of things: features, techniques and research challenges. *Int. J. Comput. Intell. Res*, v. 13, n. 7, p. 1617–1627, 2017.

HAMBY, D. M. A review of techniques for parameter sensitivity analysis of environmental models. *Environmental monitoring and assessment*, Springer, v. 32, n. 2, p. 135–154, 1994.

HAYES, B. *Cloud computing*. [S.l.]: ACM New York, NY, USA, 2008.

HEADQUARTERS, A. Cisco data center infrastructure 2.5 design guide. *Cisco Validated Design I*, Citeseer, 2007.

HILLIS, D. M.; BULL, J. J. An empirical test of bootstrapping as a method for assessing confidence in phylogenetic analysis. *Systematic biology*, Society of Systematic Biologists, v. 42, n. 2, p. 182–192, 1993.

IV, W. P. T.; PE, J.; SEADER, P.; BRILL, K. Tier classification define site infrastructure performance. *Uptime Institute*, v. 17, 2006.

KIM, M. C. Reliability block diagram with general gates and its application to system reliability analysis. *Annals of Nuclear Energy*, Elsevier, v. 38, n. 11, p. 2456–2461, 2011.

KLEYMAN, B. *Important Geographic and Risk Mitigation Factors in Selecting a Data Center Site*. 2013. Disponível em: <https://www.datacenterknowledge.com/archives/2013/07/10/important-geographic-and-risk-mitigation-factors-in-selecting-a-datacenter-site>. Acesso em: Last accessed 23 Set 2019.

KOPITTKE, H. B.; FILHO, N. C. Análise de investimentos. *São Paulo: Atlas*, 2000.

KUO, W.; ZUO, M. J. *Optimal reliability modeling: principles and applications*. [S.l.]: John Wiley & Sons, 2003.

LAPRIE, J.-C. Dependable computing and fault-tolerance. *Digest of Papers FTCS-15*, v. 10, n. 2, p. 124, 1985.

LEE, A. S. A scientific methodology for mis case studies. *MIS quarterly*, JSTOR, p. 33–50, 1989.

LENHART, T.; ECKHARDT, K.; FOHRER, N.; FREDE, H.-G. Comparison of two different approaches of sensitivity analysis. *Physics and Chemistry of the Earth, Parts A/B/C*, Elsevier, v. 27, n. 9-10, p. 645–654, 2002.

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*, Elsevier, v. 75, p. 271–282, 2017.

LIU, T.; SONG, H. Dependability prediction of high availability oscar cluster server. In: CITESEER. *Proceedings of the 2003 Int. Conf. on parallel and distributed processing techniques and applications*. [S.l.], 2003.

MACIEL, P.; BARROS, E.; ROSENSTIEL, W. A petri net model for hardware/software codesign. *Design Automation for Embedded Systems*, Springer, v. 4, n. 4, p. 243–310, 1999.

MACIEL, P.; TRIVEDI, K.; JR, R. M.; KIM, D. Performance and dependability in service computing. In: _____. [S.l.]: Igi Global, 2012. p. 45. ISBN 9781609607951.

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. Modeling availability impact in cloud computing. In: _____. *Principles of Performance and Reliability Modeling and Evaluation: Essays in Honor of Kishor Trivedi on his 70th Birthday*. Cham: Springer International Publishing, 2016. p. 287–320. ISBN 978-3-319-30599-8. Disponível em: <https://doi.org/10.1007/978-3-319-30599-8_11>.

MACIEL, P. R. M. *Performance, Reliability, and Availability Evaluation of Computational Systems, Volume 2: Reliability, Availability Modeling, Measuring, and Data Analysis*. [S.l.]: Chapman and Hall/CRC, 2022.

MACIEL, P. R. M. *Performance, Reliability, and Availability Evaluation of Computational Systems, Volume I: Performance and Background*. [S.l.]: Chapman and Hall/CRC, 2022.

MACIEL, P. R. M.; DANTAS, J. R.; JÚNIOR, R. d. S. M. Markov chains and stochastic petri nets for availability and reliability modeling. In: *Reliability Engineering*. [S.l.]: CRC Press, 2019. p. 127–151.

MANVI, S. S.; SHYAM, G. K. Resource management for infrastructure as a service (iaas) in cloud computing: A survey. *Journal of network and computer applications*, Elsevier, v. 41, p. 424–440, 2014.

MARSAN, M. A.; BALBO, G.; CONTE, G.; DONATELLI, S.; FRANCESCHINIS, G. Modelling with generalized stochastic petri nets. *ACM SIGMETRICS Performance Evaluation Review*, ACM New York, NY, USA, v. 26, n. 2, p. 2, 1998.

MARSAN, M. A.; CHIOLA, G. On petri nets with deterministic and exponentially distributed firing times. In: SPRINGER. *European Workshop on Applications and Theory in Petri Nets*. [S.l.], 1986. p. 132–145.

MARSAN, M. A.; CONTE, G.; BALBO, G. A class of generalized stochastic petri nets for the performance evaluation of multiprocessor systems. *ACM Transactions on Computer Systems (TOCS)*, ACM New York, NY, USA, v. 2, n. 2, p. 93–122, 1984.

MATOS, R.; ARAUJO, J.; OLIVEIRA, D.; MACIEL, P.; TRIVEDI, K. Sensitivity analysis of a hierarchical model of mobile cloud computing. *Simulation Modelling Practice and Theory*, Elsevier, v. 50, p. 151–164, 2015.

MATOS, R.; DANTAS, J.; ARAUJO, J.; TRIVEDI, K. S.; MACIEL, P. Redundant eucalyptus private clouds: availability modeling and sensitivity analysis. *Journal of Grid Computing*, Springer, v. 15, n. 1, p. 1–22, 2017.

MELL, P.; GRANCE, T. *The NIST Definition of Cloud Computing*. [S.l.]: Special Publication (NIST SP), National Institute of Standards and Technology, Gaithersburg, MD, 2011.

MELL, P.; GRANCE, T. The nist definition of cloud computing (nist special publication 800-145). *National Institute of Standards and Technology, Tech. Rep*, 2011.

MELO, C.; MATOS, R.; DANTAS, J.; MACIEL, P. Capacity-oriented availability model for resources estimation on private cloud infrastructure. In: IEEE. *2017 IEEE 22nd Pacific Rim International Symposium on Dependable Computing (PRDC)*. [S.l.], 2017. p. 255–260.

MELO, R.; PAULO, F. Vicente de; FILHO, I. J. de M.; FELICIANO, F.; MACIEL, P. R. M. Redundancy mechanisms applied in cloud computing infrastructures. In: IEEE. *2018 13th Iberian Conference on Information Systems and Technologies (CISTI)*. [S.l.], 2018. p. 1–6.

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. *2018 IEEE 32nd International Conference on Advanced Information Networking and Applications (AINA)*. [S.l.], 2018. p. 135–142.

MESBAHI, M. R.; RAHMANI, A. M.; HOSSEINZADEH, M. Reliability and high availability in cloud computing environments: a reference roadmap. *Human-centric Computing and Information Sciences*, Springer, v. 8, n. 1, p. 1–31, 2018.

MOLLOY, M. *On the Integration of Delay and Throughput Measures in Distributed Processing Models*. Tese (Doutorado) — University of California, Los Angeles, 01 1981.

MOLLOY, M. K. Performance analysis using stochastic petri nets. *IEEE Computer Architecture Letters*, IEEE Computer Society, v. 31, n. 09, p. 913–917, 1982.

MSEDDI, A.; SALAHUDDIN, M. A.; ZHANI, M. F.; ELBIAZE, H.; GLITHO, R. H. Efficient replica migration scheme for distributed cloud storage systems. *IEEE Transactions on Cloud Computing*, IEEE, v. 9, n. 1, p. 155–167, 2018.

MURATA, T. Petri nets: Properties, analysis and applications. *Proceedings of the IEEE*, IEEE, v. 77, n. 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, v. 56, p. 27–50, 2016.

PAN, Y.; HU, N. Research on dependability of cloud computing systems. In: IEEE. *2014 10th International Conference on Reliability, Maintainability and Safety (ICRMS)*. [S.l.], 2014. p. 435–439.

PEREIRA, P.; ARAUJO, J.; TORQUATO, M.; DANTAS, J.; MELO, C.; MACIEL, P. Stochastic performance model for web server capacity planning in fog computing. *The Journal of Supercomputing*, Springer, v. 76, n. 12, p. 9533–9557, 2020.

PETERSON, J. L. Petri nets. *ACM Computing Surveys (CSUR)*, ACM New York, NY, USA, v. 9, n. 3, p. 223–252, 1977.

PETERSON, J. L. *Petri net theory and the modeling of systems*. [S.l.]: Prentice Hall PTR, 1981.

PETRI, C. A. *Communication with automata*. Tese (Doutorado) — Universität Hamburg, 1966.

PINHEIRO, T.; OLIVEIRA, D.; MATOS, R.; SILVA, B.; PEREIRA, P.; MELO, C.; OLIVEIRA, F.; TAVARES, E.; DANTAS, J.; MACIEL, P. The mercury environment: A modeling tool for performance and dependability evaluation. In: *Intelligent Environments 2021*. [S.l.]: IOS Press, 2021. p. 16–25.

RAHMAN, A.; LIU, X.; KONG, F. A survey on geographic load balancing based data center power management in the smart grid environment. *IEEE Communications Surveys & Tutorials*, IEEE, v. 16, n. 1, p. 214–233, 2013.

RAHMAN, M. N.; ESMAILPOUR, A. A hybrid data center architecture for big data. *Big Data Research*, v. 3, p. 29–40, 2016. ISSN 2214-5796. Special Issue on Big Data from Networking Perspective. Disponível em: <https://www.sciencedirect.com/science/article/pii/S2214579616300120>.

REDHAT. *Iaas x Paas x SaaS*. 2022. Disponível em: <https://www.redhat.com/pt-br/topics/cloud-computing/iaas-vs-paas-vs-saas>. Acesso em: Last accessed 01 Jan 2022.

ROSENDO, D.; GOMES, D.; SANTOS, G. L.; GONCALVES, G.; MOREIRA, A.; FERREIRA, L.; ENDO, P. T.; KELNER, J.; SADOK, D.; MEHTA, A. et al. A methodology to assess the availability of next-generation data centers. *The Journal of Supercomputing*, Springer, v. 75, n. 10, p. 6361–6385, 2019.

ROSENDO, D.; LEONI, G.; GOMES, D.; MOREIRA, A.; GONÇALVES, G.; ENDO, P.; KELNER, J.; SADOK, D.; MAHLOO, M. How to improve cloud services availability? investigating the impact of power and it subsystems failures. In: *Proceedings of the 51st Hawaii international conference on system sciences*. [S.l.: s.n.], 2018.

SANTOS, G. L.; ENDO, P. T.; GONÇALVES, G.; ROSENDO, D.; GOMES, D.; KELNER, J.; SADOK, D.; MAHLOO, M. Analyzing the it subsystem failure impact on availability of cloud services. In: IEEE. *2017 IEEE symposium on computers and communications (ISCC)*. [S.l.], 2017. p. 717–723.

SCHEER, G. W.; DOLEZILEK, D. J. Comparing the reliability of ethernet network topologies in substation control and monitoring networks. In: *Western Power Delivery Automation Conference, Spokane, Washington*. [S.l.: s.n.], 2000.

SCHNEEWEISS, W. G. *Boolean functions: with engineering applications and computer programs*. [S.l.]: Springer Science & Business Media, 2012.

SCURRA. *tpos de nuvens*. 2022. Disponível em: <https://www.scurra.com.br/blog/infraestrutura-e-servicos-para-cloud-sobre-nuvem-hibrida/>. Acesso em: Last accessed 71 Mar 2022.

SILVA, B.; MATOS, R.; TAVARES, E.; MACIEL, P.; ZIMMERMANN, A. Sensitivity analysis of an availability model for disaster tolerant cloud computing system. *International Journal of Network Management*, Wiley Online Library, v. 28, n. 6, p. e2040, 2018.

SMITH, D. J. *Reliability, maintainability and risk: practical methods for engineers*. [S.l.]: Butterworth-Heinemann, 2021.

SOUSA, E.; LINS, F.; TAVARES, E.; MACIEL, P. Cloud infrastructure planning considering different redundancy mechanisms. *Computing*, Springer, v. 99, n. 9, p. 841–864, 2017.

TIA, A. g. c. *ANSI/TIA-942 (Telecommunications Infrastructure Standard for Data Centers)*. 2021. Disponível em: <https://www.tic.ir/Content/media/article/TIA\%20942\%20-A(2012)_0.PDF>. Acesso em: Last accessed 09 Set 2021.

TORQUATO, M.; GUEDES, E.; MACIEL, P.; VIEIRA, M. A hierarchical model for virtualized data center availability evaluation. In: IEEE. *2019 15th European Dependable Computing Conference (EDCC)*. [S.l.], 2019. p. 103–110.

TORQUATO, M.; UMESH, I.; MACIEL, P. Models for availability and power consumption evaluation of a private cloud with vmm rejuvenation enabled by vm live migration. *The Journal of Supercomputing*, Springer, v. 74, n. 9, p. 4817–4841, 2018.

UPTIME, I. L. *Uptime Institute*. 2021. Disponível em: <https://uptimeinstitute.com/>. Acesso em: Last accessed 09 Set 2021.

VERAS, M. Datacenter: componente central da infraestrutura de ti. *Rio de Janeiro: Brasport*, 2009.

WALTERS, J. P.; CHAUDHARY, V.; CHA, M.; GUERCIO, S.; GALLO, S. A comparison of virtualization technologies for hpc. In: IEEE. *22nd International Conference on Advanced Information Networking and Applications (aina 2008)*. [S.l.], 2008. p. 861–868.

WANG, D.; TRIVEDI, K. S. Computing steady-state mean time to failure for non-coherent repairable systems. *IEEE Transactions on reliability*, IEEE, v. 54, n. 3, p. 506–516, 2005.

WANG, W.; LOMAN, J. M.; ARNO, R. G.; VASSILIOU, P.; FURLONG, E. R.; OGDEN, D. Reliability block diagram simulation techniques applied to the ieee std. 493 standard network. *IEEE Transactions on Industry Applications*, IEEE, v. 40, n. 3, p. 887–895, 2004.

YEOW, W.-L.; WESTPHAL, C.; KOZAT, U. C.; KOZAT, U. C. A resilient architecture for automated fault tolerance in virtualized data centers. In: IEEE. *2010 IEEE Network Operations and Management Symposium-NOMS 2010*. [S.l.], 2010. p. 866–869.

ZHANG, Q.; CHENG, L.; BOUTABA, R. Cloud computing: state-of-the-art and research challenges. *Journal of internet services and applications*, SpringerOpen, v. 1, n. 1, p. 7–18, 2010.

APÊNDICE A – TEMPOS DE FALHA E REPARO DAS CAMADAS

Aqui estão apresentados os tempos de falha e reparo das camadas de balanceador de carga, aplicação e Banco de dados. São apresentados, também, os valores de TTF's e TTR's do DCS.

A.1 CAMADA DE BALANCEADOR DE CARGA

Nesta seção está apresentado os TTF's e TTR's dos componentes da camada de balanceador de carga, VM e Apache. Os valores aqui citados foram obtidos através do monitoramento que perdurou cerca de seis meses. Sempre que houve uma alteração de estado de cada componente, foram registrados os TTF's e TTR's.

CAMADA DE BALANCEADOR DE CARGA

VM

Nº	TTF (h)	TTR (h)	Nº	TTF (h)	TTR (h)
1	27,30	0,35	23	23,45	0,03
2	0,08	0,03	24	74,95	0,14
3	168	0,17	25	142,32	0,05
4	25	0,05	26	184,59	0,12
5	8	0,17	27	102,79	0,18
6	81	0,17	28	117,34	0,03
7	240,20	0,03	29	103,07	0,03
8	196,25	0,03	30	77,61	0,43
9	0,65	0,43	31	106,78	0,17
10	0,45	0,17	32	79,68	0,10
11	0,12	0,10	33	201,70	0,17
12	294,60	0,17	34	91,89	0,10
13	0,03	0,10	35	112,32	0,03
14	54,20	0,03	36	45,65	0,04
15	0,08	0,03	37	15,57	0,15
16	4,12	0,05	38	64,20	0,17
17	29,67	0,17	39	46,71	0,24
18	264,00	0,17	40	149,64	0,03
19	343,65	0,03	41	72,08	0,13
20	307,01	0,13	42	132,30	0,13
21	15,93	0,08	43	52,00	0,03
22	103,86	0,35	44	158,56	0,03

APACHE

Nº	TTF (h)	TTR (h)
1	7,80	0,15
2	896,61	0,50
3	2196,00	0,67
4	1034,60	0,44

A.2 CAMADA DE APLICAÇÃO

Nesta seção está apresentado os TTF's e TTR's dos componentes da camada de aplicação, VM e Tomcat. Os valores aqui apresentados também foram obtidos através do monitoramento que perdurou cerca de seis meses.

CAMADA DE APLICAÇÃO

VM

Nº	TTF (h)	TTR (h)
1	8	0,17
2	1299	0,17
3	37,37	0,17
4	9,82	0,03
5	63,20	0,17
6	125,98	0,03
7	685,32	0,17
8	1045,52	0,03
9	652,30	0,17
10	63,30	0,17
11	85,52	0,17
12	95,63	0,17
13	224,20	0,17

TOMCAT

Nº	TTF (h)	TTR (h)	Nº	TTF (h)	TTR (h)	Nº	TTF (h)	TTR (h)
1	5,43	0,08	44	3,35	0,08	87	77,36	0,41
2	2,01	0,18	45	96,97	0,08	88	83,91	0,32
3	2,43	0,08	46	0,59	0,08	89	43,70	0,40
4	5,12	0,08	47	0,26	0,08	90	44,92	0,04
5	7,74	0,08	48	0,68	0,08	91	10,63	0,02
6	11,25	0,08	49	0,09	0,08	92	18,20	0,03
7	1,76	0,08	50	0,09	0,08	93	34,51	0,01
8	30,35	0,72	51	0,43	0,08	94	21,20	0,08
9	27,27	0,08	52	0,84	0,08	95	51,70	0,08
10	76,53	0,08	53	0,18	0,08	96	8,00	0,08
11	48,67	0,08	54	0,34	0,08	97	45,41	0,08
12	28,05	0,08	55	0,59	0,08	98	0,15	0,04
13	15,44	0,08	56	0,43	0,08	99	38,00	0,08
14	0,27	0,46	57	24,03	0,08	100	86,18	0,08
15	104,60	0,09	58	0,09	0,08	101	3,41	0,08
16	86,02	0,08	59	45,43	0,18	102	155,81	0,01
17	8,12	0,08	60	2,75	0,08	103	38,70	0,08
18	14,70	0,08	61	23,92	0,08	104	5,91	0,28
19	4,76	0,08	62	36,12	0,08	105	34,89	0,08
20	112,52	0,08	63	91,70	0,08	106	21,76	0,01
21	7,78	0,08	64	159,02	0,08	107	2,87	0,01
22	0,18	0,08	65	29,82	0,08	108	42,12	0,64
23	23,06	0,08	66	70,90	0,12	109	30,88	0,04
24	39,01	0,08	67	32,10	0,08	110	26,31	0,45
25	179,77	0,08	68	26	0,08	111	21,10	0,17
26	21,18	0,08	69	45,52	0,63	112	24,79	0,12
27	39,01	0,08	70	61,13	0,08	113	36,43	0,33
28	0,09	0,92	71	12,20	0,01	114	27,20	0,02
29	7,07	0,08	72	45,51	0,25	115	1,23	0,01
30	0,18	0,27	73	35,57	0,01	116	28,85	0,04
31	13,47	1,18	74	86,29	0,23	117	4,32	0,01
32	25,10	0,08	75	9,12	0,01	118	32,99	0,24
33	149,72	0,08	76	17,45	0,01	119	29,13	0,83
34	101,53	0,08	77	3,24	0,01	120	15,38	0,19
35	36,05	0,08	78	15,51	0,01	121	12,20	0,46
36	54,78	0,08	79	9,30	0,01	122	7,94	0,08
37	14,10	0,08	80	9,17	0,01	123	49,26	0,08
38	17,41	0,08	81	97,56	0,08	124	41,98	0,08
39	167,02	0,08	82	58,81	0,08	125	53,87	0,07
40	79,43	0,08	83	31,02	0,08	126	41,23	0,01
41	98,82	0,08	84	23,98	0,08	127	23,80	0,24
42	22,60	0,08	85	4,46	0,01			
43	48,08	0,08	86	30,82	0,23			

A.3 CAMADA DE BANCO DE DADOS

Nesta seção está apresentado os TTF's e TTR's dos componentes da camada de banco de dados, VM e Oracle. Os valores aqui apresentados também foram obtidos através do monitoramento que perdurou cerca de seis meses.

CAMADA DE BANCO DE DADOS								
VM						ORACLE		
Nº	TTF (h)	TTR (h)	Nº	TTF (h)	TTR (h)	Nº	TTF (h)	TTR (h)
1	14,90	0,33	23	239,67	0,03	1	0,03	0,89
2	25,17	0,03	24	1,17	0,17	2	169,01	0,01
3	148,62	0,03	25	360,33	0,17	3	0,03	0,01
4	90,12	0,05	26	22,73	0,03	4	0,00	0,01
5	24	0,03	27	1,00	0,03	5	643,82	0,42
6	48	0,03	28	72,87	0,03	6	7,11	1
7	24	0,03	29	129,62	0,03	7	936,00	1,45
8	24,00	0,03	30	0,03	0,03	8	1,03	0,05
9	0,75	0,03	31	0,05	0,03	9	360,43	7,01
10	47,45	0,03	32	61,17	0,05	10	35,05	0,05
11	24,37	0,03	33	0,03	0,03	11	0,54	0,05
12	23,35	0,03	34	0,03	0,03	12	1104,17	0,05
13	0,63	0,03	35	34,00	0,03	13	26,33	0,97
14	23,48	0,03	36	57,58	0,17	14	896,45	0,75
15	25,70	0,03	37	43,20	0,03	15	41,23	0,99
16	46,05	0,03	38	343,65	0,03	16	124,52	0,90
17	0,95	0,03	39	127,80	0,01			
18	23,08	0,05	40	179,85	0,03			
19	791,00	0,03	41	15,93	0,08			
20	8	0,17	42	33,30	0,03			
21	937	0,17	43	98,43	0,03			
22	120,27	0,03						

A.4 ESTRUTURA DO CENTRO DE DADOS

Nesta seção estão registrados os TTF's e TTR's do Centro de Dados, assim como os valoroes de MTTF, MTTR e disponibilidade do CD. Estes valores também foram retirados do monitoramento, porém, diferente dos demais, o período de checagem foi de aproximadamente 9500h.

ESTRUTURA DO CENTRO DE DADOS		
Nº	**TTF (h)**	**TTR (h)**
1	2510	4,7
2	2436	7,3
3	1582	5,8
4	2933	4,2

Métrica	**Valor**
Tempo de análise (h)	9483
MTTF (h)	2365,25
MTTR (h)	5,5
A	0,9976801
#9s	2,6345231
DT (h)	1219,36

APÊNDICE B – INFORMAÇÕES UTILIZADAS NOS MODELOS

Aqui estão apresentados em forma de tabela as informações utilizadas nos modelos apresentados neste trabalho. Estão apresentados informações referentes aos modelos em RBD e SPN, que representam a modelagem hierárquica.

B.1 CAMADA DE CONECTIVIDADE E ARMAZENAMENTO

Nesta seção está apresentado os parâmetros utilizados nos modelos em SPN das camadas de conectividade e armazenamento.

CAMADA DE CONECTIVIDADE				
Item	**Tipo**	**Valor (h)**	**Expressão de guarda**	**Descrição**
NET_UP	Lugar	N/A	N/A	Camada de conectividade no estado operacional
NET_DW	Lugar	N/A	N/A	Camada de conectividade no estado de defeito
F_NET	Transição temporizada	87600	N/A	MTTF da camada de conectividade
R_NET	Transição temporizada	24	N/A	MTTR da camada de conectividade

CAMADA DE ARMAZENAMENTO				
SWITCH ETHERNET				
Item	**Tipo**	**Valor (h)**	**Expressão de guarda**	**Descrição**
SAN_UP	Lugar	N/A	N/A	Switch SAN no estado operacional
SAN_DW	Lugar	N/A	N/A	Switch SAN no estado de defeito
F_SAN	Transição temporizada	87600	N/A	MTTF do Switch SAN
R_SAN	Transição temporizada	48	N/A	MTTR do Switch SAN
UNIDADE DE ARMAZENAMENTO				
Item	**Tipo**	**Valor (h)**	**Expressão de guarda**	**Descrição**
STG_UP	Lugar	N/A	N/A	Unidade de armazenamento no estado operacional
STG_DW	Lugar	N/A	N/A	Unidade de armazenamento no estado de defeito
F_STG	Transição temporizada	131400	N/A	MTTF da unidade de armazenamento
R_STG	Transição temporizada	4	N/A	MTTR da unidade de armazenamento

B.2 CAMADA DE VIRTUALIZAÇÃO

Nesta seção está apresentado os parâmetros utilizados nos modelos em RBD e SPN da camada de virtualização.

CAMADA DE VIRTUALIZAÇÃO – 1º NÍVEL HIERÁRQUICO (RBD)				
Item	**Tipo**	**MTTF (h)**	**MTTR (h)**	**Descrição**
HW	Bloco	26280	24	Componentes físicos do servidor
HP	Bloco	8760	1,67	SO de virtualização (Hipervisor)

CAMADA DE VIRTUALIZAÇÃO – 2º NÍVEL HIERÁRQUICO (SPN)				
SERVIDOR SEM REDUNDÂNCIA OU EM HA				
Item	**Tipo**	**Valor (h)**	**Expressão de guarda**	**Descrição**
SRV_UP	Lugar	N/A	N/A	Camada de virtualização no estado operacional
SRV_DW	Lugar	N/A	N/A	Camada de virtualização no estado de defeito
F_SRV	Transição temporizada	6502,26	N/A	MTTF da camada de virtualização
R_SRV	Transição temporizada	7,196	N/A	MTTR da camada de virtualização
SERVIDOR EM COLD STANDBY				
Item	**Tipo**	**Valor (h)**	**Expressão de guarda**	**Descrição**
SRV1_UP	Lugar	N/A	N/A	Servidor principal no estado operacional
SRV1_DW	Lugar	N/A	N/A	Servidor principal no estado de defeito
F_SRV1	Transição temporizada	6502,26	N/A	MTTF da servidor principal
R_SRV1	Transição temporizada	7,196	N/A	MTTR da servidor principal
SRV2_UP	Lugar	N/A	N/A	Servidor secundário no estado operacional
SRV2_DW	Lugar	N/A	N/A	Servidor secundário no estado de defeito
F_SRV2	Transição temporizada	6502,26	N/A	MTTF da Servidor secundário
R_SRV2	Transição temporizada	7,196	N/A	MTTR da servidor secundário
E_SRV	Lugar	N/A	N/A	Estado especial - servidor principal está operacional
START_SRV2	Transição temporizada	0,16	N/A	Inicialização do servidor secundário
STDW_SRV2	Transição Imediata	N/A	(#VC2_UP=0) AND (#VC2_DW=0) AND (#LB2_UP=0) AND (#LB2_DW=0)	Desligamento do servidor secundário

B.3 CAMADA DE APLICAÇÃO

Nesta seção está apresentado os parâmetros utilizados nos modelos em RBD e SPN da camada de aplicação.

CAMADA DE APLICAÇÃO – 1º NÍVEL HIERÁRQUICO (RBD)				
Item	**Tipo**	**MTTF (h)**	**MTTR (h)**	**Descrição**
SO	Bloco	338,54	0,13	Sistema Operacional da VM
APP	Bloco	34,71	0,13	Aplicação instalada no SO (Tomcat)

CAMADA DE APLICAÇÃO – 2º NÍVEL HIERÁRQUICO (SPN)				
SERVIDOR SEM REDUNDÂNCIA OU EM HA				
Item	**Tipo**	**Valor (h)**	**Expressão de guarda**	**Descrição**
APP_UP	Lugar	N/A	N/A	Camada de aplicação no estado operacional
APP_DW	Lugar	N/A	N/A	Camada de aplicação no estado de defeito
F_APP	Transição temporizada	31,482411	N/A	MTTF da camada de aplicação
R_APP	Transição temporizada	0,134118	(#SAN_UP>0) AND (#STG_UP>0 AND (#SRV2_UP>0)AND (#VC_UP>0) AND (#DB_UP>0)	MTTR da camada de aplicação
STDW_APP	Transição Imediata	N/A	(#SAN_UP=0) OR (#STG_UP=0) OR (#SRV2_UP=0)	Desligamento imediato após falhas externas
START_APP	Transição temporizada	0,08	(#SAN_UP>0) AND (#STG_UP>0) AND (#SRV2_UP>0) AND (#DB_UP>0)	Tempo de inicialização da camada de aplicação
E_APP_DW	Lugar	N/A	N/A	Estado de defeito após interferências externas

B.4 CAMADA DE BANCO DE DADOS

Nesta seção está apresentado os parâmetros utilizados nos modelos em RBD e SPN da camada de banco de dados.

CAMADA DE BANCO DE DADOS – 1º NÍVEL HIERÁRQUICO (RBD)				
Item	**Tipo**	**MTTF (h)**	**MTTR (h)**	**Descrição**
SO	Bloco	99,84	0,05	Sistema Operacional da VM
BD	Bloco	271,43	0,91	Aplicação instalada no SO (Oracle)

CAMADA DE BANCO DE DADOS – 2º NÍVEL HIERÁRQUICO (SPN)				
SERVIDOR SEM REDUNDÂNCIA OU EM HA				
Item	**Tipo**	**Valor (h)**	**Expressão de guarda**	**Descrição**
DB_UP	Lugar	N/A	N/A	Camada de banco de dados no estado operacional
DB_DW	Lugar	N/A	N/A	Camada de banco de dados no estado de defeito
F_DB	Transição temporizada	72,993804	N/A	MTTF da camada de banco de dados
R_DB	Transição temporizada	0,2837732	(#SAN_UP>0) AND (#STG_UP>0) AND (#SRV3_UP>0) AND (#VC_UP>0)	MTTR da camada de banco de dados
STDW_DB	Transição Imediata	N/A	(#SAN_UP=0) OR (#STG_UP=0) OR (#SRV3_UP=0)	Desligamento imediato após falhas externas
START_DB	Transição temporizada	0,16	(#SAN_UP>0) AND (#STG_UP>0) AND (#SRV3_UP>0)	Tempo de inicialização da camada de banco de dados
E_DB_DW	Lugar	N/A	N/A	Estado de defeito após interferências externas

B.5 CAMADA DE ORQUESTRAÇÃO

Nesta seção está apresentado os parâmetros utilizados nos modelos em RBD e SPN da camada de orquestração.

CAMADA DE ORQUESTRAÇÃO – 1º NÍVEL HIERÁRQUICO (RBD)				
Item	**Tipo**	**MTTF (h)**	**MTTR (h)**	**Descrição**
SO	Bloco	56543	8,6	Sistema Operacional da VM
VC	Bloco	46598	7,5	Aplicação instalada no SO (Vcenter)

CAMADA DE ORQUESTRAÇÃO – 2º NÍVEL HIERÁRQUICO (SPN)				
SERVIDOR SEM REDUNDÂNCIA OU EM HA				
Item	**Tipo**	**Valor (h)**	**Expressão de guarda**	**Descrição**
VC_UP	Lugar	N/A	N/A	Camada de orquestração no estado operacional
VC_DW	Lugar	N/A	N/A	Camada de orquestração no estado de defeito
F_VC	Transição temporizada	25535	N/A	MTTF da camada de orquestração
R_VC	Transição temporizada	8	(#SAN_UP>0) AND (#STG_UP>0) AND (#SRV1_UP>0)	MTTR da camada de orquestração
STDW_VC	Transição Imediata	N/A	(#SAN_UP=0) OR (#STG_UP=0) OR (#SRV1_UP=0)	Desligamento imediato após falhas externas
START_VC	Transição temporizada	0,33	(#SAN_UP>0) AND (#STG_UP>0) AND (#SRV1_UP>0)	Tempo de inicialização da camada de orquestração
E_VC_DW	Lugar	N/A	N/A	Estado de defeito após interferências externas

SERVIDOR COM REDUNDÂNCIA - COLD STANDBY				
Item	**Tipo**	**Valor (h)**	**Expressão de guarda**	**Descrição**
VC1_UP	Lugar	N/A	N/A	Camada de orquestração no estado operacional
VC1_DW	Lugar	N/A	N/A	Camada de orquestração no estado de defeito
F_VC1	Transição temporizada	25535	N/A	MTTF da camada de orquestração
R_VC1	Transição temporizada	8	(#SAN_UP>0) AND (#STG_UP>0) AND (#SRV1_UP>0)	MTTR da camada de orquestração
VC2_UP	Lugar	N/A	N/A	Camada de orquestração no estado operacional
VC2_DW	Lugar	N/A	N/A	Camada de orquestração no estado de defeito
F_VC2	Transição temporizada	25535	N/A	MTTF da camada de orquestração
R_VC2	Transição temporizada	8	(#SAN_UP>0) AND (#STG_UP>0) AND (#SRV2_UP>0)	MTTR da camada de orquestração
STDW_VC1	Transição Imediata	N/A	(#SAN_UP=0) AND (#STG_UP=0) OR (#SRV1_UP=0)	Desligamento imediato após falhas externas
STDW_VC2	Transição Imediata	N/A	(#SAN_UP=0) AND (#STG_UP=0) OR (#SRV2_UP=0)	Desligamento imediato após falhas externas
START_VC1	Transição temporizada	0,33	(#SAN_UP>0) AND (#STG_UP>0) AND (#SRV1_UP>0)	Tempo de inicialização da camada de orquestração
START_VC2	Transição temporizada	0,33	(#SAN_UP>0) AND (#STG_UP>0) AND (#SRV11_UP>0)	Tempo de inicialização da camada de orquestração
E_VC_DW	Lugar	N/A	N/A	Estado de defeito após interferências externas
MGT_VC_UP	Transição Imediata	N/A	(#SRV1_UP>0)	Migração online da VM do servidor 02 para o servidor 01
MGT_VC_DW1	Transição Imediata	N/A	(#SRV1_UP>0)	Migração a frio da VM do servirdor 01 para o servidor 02
MGT_VC_DW2	Transição Imediata	N/A	(#SRV1_UP=0) AND (#SRV2_UP>0)	Migração a frio da VM do servirdor 02 para o servidor 01

B.6 CAMADA DE BALANCEADOR DE CARGA

Nesta seção está apresentado os parâmetros utilizados nos modelos em RBD e SPN da camada de balenceador de carga.

CAMADA DE BALANCEADOR DE CARGA – 1º NÍVEL HIERÁRQUICO (RBD)				
Item	**Tipo**	**MTTF (h)**	**MTTR (h)**	**Descrição**
SO	Bloco	98,11	0,12	Sistema Operacional da VM
LB	Bloco	1033,46	0,43	Aplicação instalada no SO (Apache)

CAMADA DE BALANCEDOR DE CARGA – 2º NÍVEL HIERÁRQUICO (SPN)				
SERVIDOR SEM REDUNDÂNCIA OU EM HA				
Item	**Tipo**	**Valor (h)**	**Expressão de guarda**	**Descrição**
LB_UP	Lugar	N/A	N/A	Camada de balanceador de carga no estado operacional
LB_DW	Lugar	N/A	N/A	Camada de balanceador de carga no estado de defeito
F_LB	Transição temporizada	89,6041568	N/A	MTTF da camada de balanceador de carga
R_LB	Transição temporizada	0,15081074	(#SAN_UP>0)AND (#STG_UP>0) AND (#SRV1_UP>0) AND (#VC_UP>0)	MTTR da camada de balanceador de carga
STDW_LB	Transição Imediata	N/A	(#SAN_UP=0) OR (#STG_UP=0) OR (#SRV1_UP=0)	Desligamento imediato após falhas externas
START_LB	Transição temporizada	0,08	(#SAN_UP>0) AND (#STG_UP>0) AND (#SRV1_UP>0)	Tempo de inicialização da camada de balanceador de carga
E_LB_DW	Lugar	N/A	N/A	Estado de defeito após interferências externas

SERVIDOR COM REDUNDÂNCIA - COLD STANDBY				
Item	**Tipo**	**Valor (h)**	**Expressão de guarda**	**Descrição**
LB1_DW	Lugar	N/A	N/A	Camada de balanceador de carga no estado de defeito
F_LB1	Transição temporizada	89,6041568	N/A	MTTF da camada de balanceador de carga
R_LB1	Transição temporizada	0,15081074	(#SAN_UP>0) AND (#STG_UP>0) AND (#SRV1_UP>0) AND (#VC1_UP>0)	MTTR da camada de balanceador de carga
LB2_UP	Lugar	N/A	N/A	Camada de balanceador de carga no estado operacional
LB2_DW	Lugar	N/A	N/A	Camada de balanceador de carga no estado de defeito
F_LB2	Transição temporizada	89,6041568	N/A	MTTF da camada de balanceador de carga
R_LB2	Transição temporizada	0,15081074	(#SAN_UP>0) AND (#STG_UP>0) AND (#SRV2_UP>0) AND (#VC2_UP>0)	MTTR da camada de balanceador de carga
STDW_LB1	Transição Imediata	N/A	(#SAN_UP=0)AND(#STG_UP=0)OR(#SRV 1_UP=0)	Desligamento imediato após falhas externas
STDW_LB2	Transição Imediata	N/A	(#SAN_UP=0) AND (#STG_UP=0) OR (#SRV2_UP=0)	Desligamento imediato após falhas externas
START_LB1	Transição temporizada	0,08	(#SAN_UP>0) AND (#STG_UP>0) AND (#SRV1_UP>0)	Tempo de inicialização da camada de balanceador de carga
START_LB2	Transição temporizada	0,08	(#SAN_UP>0) AND (#STG_UP>0) AND (#SRV2_UP>0)	Tempo de inicialização da camada de balanceador de carga
E_LB_DW	Lugar	N/A	N/A	Estado de defeito após interferências externas
MGT_LB_UP	Transição Imediata	N/A	(#SRV1_UP>0)	Migração online da VM do servidor 02 para o servidor 01
MGT_LB_DW1	Transição Imediata	N/A	(#SRV1_UP>0)	Migração a frio da VM do servirdor 01 para o servidor 02
MGT_LB_DW2	Transição Imediata	N/A	(#SRV1_UP=0) AND (#SRV2_UP>0)	Migração a frio da VM do servirdor 02 para o servidor 01

B.7 ESTRUTURAS DO CENTRO DE DADOS E SISTEMA

Nesta seção está apresentado os parâmetros utilizados no modelo em SPN das estruturas do centro de dados e do sistema.

ESTRUTURAS DO CENTRO DE DADOS E DO SISTEMA				
Item	**Tipo**	**Valor (h)**	**Expressão de guarda**	**Descrição**
DC_UP	Lugar	N/A	N/A	CD no estado operacional
DC_DW	Lugar	N/A	N/A	CD no estado de defeito
DC_F	Transição temporizada	2365,25	N/A	MTTF do CD
DC_R	Transição temporizada	5,5	N/A	MTTR do CD
SS_UP	Lugar	N/A	N/A	Sistema no estado operacional
SS_DW	Lugar	N/A	N/A	Sistema no estado de defeito
SS_F	Transição temporizada	1714,735	N/A	MTTF do Sistema
SS_R	Transição temporizada	12,65708	(#DC_UP>0)	MTTR do Sistema
SS_START	Transição temporizada	0,5	N/A	Tempo de inicialização do sistema
SS_E	Lugar	N/A	N/A	Estado indicando que o sistema está indiponível por problemas no CD
SS_STDW	Transição Imediata	N/A	N/A	Desligamento imediato após falha no CD

B.8 REPLICAÇÃO DO CENTRO DE DADOS

Nesta seção está apresentado os parâmetros utilizados no modelo em SPN da estrutura de replicação do centro de dados.

REPLICAÇÃO DO CENTRO DE DADOS				
CONTROLE DO CENTRO DE DADOS				
Item	**Tipo**	**Valor (h)**	**Expressão de guarda**	**Descrição**
P_A	Lugar	N/A	N/A	CD principal no estado ativo
DC_DW	Lugar	N/A	N/A	Estado temporário de um CD após falha
S_A	Lugar	N/A	N/A	CD secundário no estado ativo
P_START	Transição Imediata	N/A	(#PDC_UP>#P_A)	Redirecionar o tráfego externo para o CD principal
P_STDW	Transição Imediata	N/A	(#PDC_UP<#P_A)	Retirando o tráfego externo do CD principal
S_START	Transição Imediata	N/A	(#SDC_UP=1) AND (#S_A=0)	Redirecionar o tráfego externo para o CD secundário
S_STDW	Transição Imediata	N/A	(#SDC_UP=0)	Retirando o tráfego externo do CD secundário
CENTRO DE DADOS PRINCIPAL				
Item	**Tipo**	**Valor (h)**	**Expressão de guarda**	**Descrição**
PDC_UP	Lugar	N/A	N/A	CD principal no estado operacional
PDC_DW	Lugar	N/A	N/A	CD principal no estado de defeito
PDC_F	Transição temporizada	9683,901	N/A	MTTF do CD principal
PDC_R	Transição temporizada	25,636947	N/A	MTTR do CD principal
CCF1	Transição temporizada	8760	N/A	MTTF da Causa comum de falha
CCR1	Transição temporizada	24	N/A	MTTR da Causa comum de Reparo
PE_DW	Lugar	N/A	N/A	Estado indicando que o CD principal está indiponível por um CCF
CENTRO DE DADOS SECUNDÁRIO				
Item	**Tipo**	**Valor (h)**	**Expressão de guarda**	**Descrição**
SDC_UP	Lugar	N/A	N/A	CD secundário no estado operacional
SDC_DW	Lugar	N/A	N/A	CD secundário no estado de defeito
SDC_F	Transição temporizada	9683,901	N/A	MTTF do CD secundário
SDC_R	Transição temporizada	25,636947	N/A	MTTR do CD secundário
CCF2	Transição temporizada	8760	N/A	MTTF da Causa comum de falha
CCR2	Transição temporizada	24	N/A	MTTR da Causa comum de Reparo
SE_DW	Lugar	N/A	N/A	Estado indicando que o CD secundário está indiponível por um CCF