

# Modelo Estocástico para Avaliação de Disponibilidade de Hospitais Inteligentes

Laécio Rodrigues<sup>±</sup>, Igor Gonçalves<sup>±</sup>, Iure Fé<sup>\*</sup>, Patricia Takako Endo<sup>⊗</sup>,  
Francisco Airton Silva<sup>±</sup>

<sup>±</sup> Universidade Federal do Piauí (UFPI) - Picos, PI, Brasil

<sup>\*</sup> 3º BEC - Exército Brasileiro - Picos, PI, Brasil

<sup>⊗</sup> Universidade de Pernambuco, Recife, PE, Brasil

faps@ufpi.edu.br, patricia.endo@upe.br

**Abstract.** *Smart hospitals need local and remote servers to efficiently process and store data. However, there is a significant difficulty in assessing the availability of such systems in real contexts, because failures are not tolerated and the cost of prototyping is high. This paper adopts Stochastic Petri Nets (SPNs) to assess the availability of an smart hospital system, avoiding premature investment in real equipment. In addition, this work presents a sensitivity analysis that identifies the most critical architecture components. The proposed model has the potential to assist hospital systems administrators in planning more optimized architectures according to their needs.*

**Resumo.** *Hospitais inteligentes precisam de servidores locais e remotos para processar e armazenar dados com eficiência. No entanto, há uma dificuldade significativa na avaliação da disponibilidade de tais sistemas em contextos reais, porque as falhas não são toleradas e o custo de prototipação é alto. Este artigo adota Redes de Petri Estocásticas (SPNs) para avaliar a disponibilidade de um sistema hospitalar inteligente, evitando o investimento prematuro em equipamentos reais. Além disso, este trabalho apresenta uma análise de sensibilidade que identifica os componentes mais críticos da arquitetura. Este trabalho tem o potencial de auxiliar administradores de sistemas hospitalares no planejamento de arquiteturas mais otimizadas de acordo com suas necessidades.*

## 1. Introdução

De acordo com [Farahani et al. 2020], a interação entre tecnologia e saúde tem uma longa história. No entanto, nos últimos anos, com o advento da Internet das Coisas (do inglês *Internet of Things* - IoT), dos biossensores vestíveis e dos avanços em técnicas de *big data*, novas oportunidades na área de serviços de saúde personalizados foram criadas para manipulação eficaz de dados multiescalares, multimodais e distribuídos. Tecnologias IoT podem ser utilizadas por pessoas de diversas idades, mas têm sido amplamente adotada por pessoas mais idosas [Kim 2018]. De acordo com um relatório publicado pela Deloitte Insights<sup>1</sup>, a prestação de serviços focados aos cuidados de saúde à população geriátrica é uma preocupação relevante para governos e sistemas de saúde. Os gastos do mercado global com cuidados geriátricos (*home care*, monitoramento remoto de pacientes, entre outros) provavelmente excederão 1,4 trilhão de dólares em 2023.

<sup>1</sup>Disponível em: <https://documents.deloitte.com/insights/2020globalhealthcareoutlook>

Contudo, apesar de todo o investimento financeiro planejado, a crise de saúde causada pela pandemia do COVID-19 tem ressaltado a fragilidade dos sistemas de saúde ao redor do mundo e a necessidade de otimizar a utilização dos (escassos) recursos existentes. Para se preparar para uma pandemia como esta que estamos enfrentando, os hospitais precisam de uma estratégia para gerenciar seu espaço, equipe e suprimentos, a fim de proporcionar o melhor atendimento aos pacientes [Wong et al. 2020]. A utilização da IoT em hospitais inteligentes se destacou por permitir a aplicação de novas metodologias de monitoramento de pacientes, com coleta de dados mais precisos e com mais riqueza de informações para auxiliar no processo de tomada de decisão [Rahmani et al. 2015, Rahmani et al. 2018, Zhang et al. 2018].

Neste contexto, os dados vitais dos pacientes podem ser capturados não apenas por um sensor, mas por uma rede de sensores sem fio (do inglês *wireless sensor networks* - WSNs), composta por um conjunto de sensores autônomos distribuídos, que transmitem cooperativamente seus dados. Apesar das vantagens das WSNs, como escalabilidade e mobilidade, esses sistemas podem enfrentar falhas em diversos pontos, comprometendo assim o monitoramento de diversos pacientes [Gomez-Sacristan et al. 2015]. Assim, torna-se relevante e necessário avaliar a disponibilidade das arquiteturas de computação (WSN e demais pontos de computação, como processamento e armazenamento de dados) em hospitais inteligentes, mesmo antes de implementar uma infraestrutura real.

Dentre as diversas metodologias existentes, as redes de Petri estocásticas (do inglês, *Stochastic Petri Nets* - SPNs) proporcionam a criação de modelos analíticos que representam sistemas complexos com características diversas, incluindo paralelismo e simultaneidade, podendo ser utilizadas para avaliação de disponibilidade de sistemas [Silva et al. 2017, Silva et al. 2015b]. O presente artigo parte de uma arquitetura de hospital inteligente previamente proposta por [Rahmani et al. 2015] para análise de disponibilidade e sensibilidade para detectar componentes com maior impacto na disponibilidade geral do sistema em questão. Este trabalho estende nosso trabalho anterior [Rodrigues et al. 2019], que focou na avaliação de métricas de desempenho, como tempo de resposta e utilização de recursos. Assim, as principais contribuições deste artigo são: (i) Um modelo SPN para avaliação de disponibilidade de hospitais inteligentes, com a capacidade de configuração de tempos médios de falha e de recuperação de 16 componentes; (ii) Exploração de diferentes cenários para avaliação de disponibilidade; e (iii) Análise de sensibilidade sobre os componentes da arquitetura de um hospital inteligente.

Alguns artigos relacionados abordaram modelos SPN para avaliar sistemas computacionais em saúde com apoio de sensores IoT [Pantelopoulos and Bourbakis 2009], [Araujo et al. 2014], [Santos et al. 2018a], [Tigre et al. 2018], [Santos et al. 2018b]. Dentre os trabalhos citados, apenas [Tigre et al. 2018] e [Santos et al. 2018b] aplicaram análise de sensibilidade sobre os modelos propostos. No entanto, nosso trabalho além da análise de sensibilidade explora a dependência dos componentes da arquitetura, adicionando a técnica de falha em cascata [Araujo et al. 2019], o que permite maior acuidade dos resultados.

O restante do artigo está organizado da seguinte forma. A Seção 2 apresenta os conceitos base para entendimento do trabalho. A Seção 3 mostra uma visão geral da arquitetura avaliada. A Seção 4 apresenta modelo SPN proposto. A Seção 5 detalha um estudo de caso e a análise de sensibilidade. A Seção 6 traça algumas conclusões.

## 2. Background

Esta seção descreve conceitos fundamentais para o entendimento deste trabalho.

### 2.1. Redes de Petri Estocásticas

As Redes de Petri (do inglês *Petri Nets* - PNs) são ferramentas de modelagem gráfica e matemática usadas para representar diferentes tipos de sistemas, caracterizados por simultaneidade, assincronia, distribuição, paralelismo, não determinismo, e/ou processos estocásticos [Chen and Ha 2018]. As SPNs (Redes de Petri Estocásticas) são uma das classes de PNs mais comumente usadas. SPNs são PNs temporizadas, significando que a noção de tempo é incorporada na semântica da rede. No caso de SPNs, o atraso de disparo está associado às transições, e as transições são disparadas com atraso de tempo determinado a partir de uma distribuição aleatória definida, como a distribuição exponencial [Xiaoli et al. 2007].

As SPNs podem ser identificadas como um tipo de grafo direcionado dividido em duas partes, preenchido por três tipos de objetos. Esses objetos são lugares, transições e arcos direcionados que conectam os lugares para as transições e transições aos lugares. A Figura 1 ilustra dois tipos de transições (temporizadas e imediatas). A transição temporizada segue um comportamento estocástico, em conformidade com uma função de distribuição de probabilidade. A transição imediata dispara quando está ativado, sem esperar nenhum período de tempo. O círculo branco simboliza lugares. Os arcos são usados para conectar os lugares as transições. Os arcos inibidores bloqueiam ou permitem a passagem de tokens de um lugar para outro, e o token é atribuído a um determinado lugar.

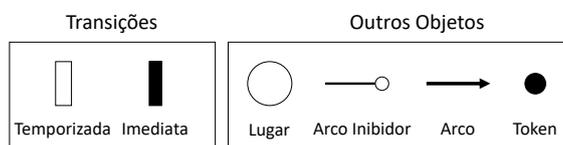


Figura 1. Componentes de um Modelo SPN

### 2.2. Análise de Sensibilidade

A análise de sensibilidade é uma medida do efeito local de um determinado dado de entrada em relação ao dado de saída, tendo como objetivo esquematizar os elos fracos dos sistemas computacionais, e daí em diante, buscar adotar um conjunto de técnicas que visam melhorias para esses sistemas em diferentes cenários [Campolongo et al. 1999]. De certo modo, a análise de sensibilidade pode trazer uma segurança necessária, e encaminhar os resultados dentro da perspectiva pré-estabelecida pelos administradores do sistema.

Existem muitas maneiras de executar a análise de sensibilidade. Vários métodos estão à disposição, como: análise de regressão, análise de perturbação (PA), variação um por um, simulação de Monte Carlo, análise diferencial paramétrica, análise de correlação e diferença percentual. A distinção de qual método utilizar é uma etapa difícil, se fazendo necessário estabelecer quais recursos computacionais são disponíveis, e as características dos problemas abordados [Campolongo et al. 2004] [Pianosi et al. 2016].

O método de diferença percentual foi escolhido para realizar a análise de sensibilidade deste trabalho, sendo uma abordagem que não utiliza um domínio contínuo para

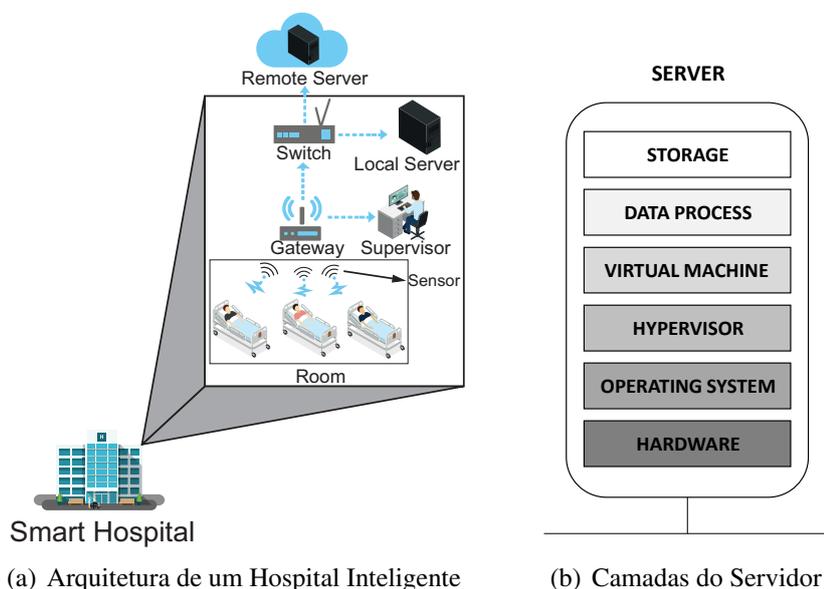
os valores de entrada de parâmetros. Este processo se baseia no cálculo da diferença percentual, ao variar um parâmetro de entrada de valor mínimo para o seu valor máximo. É necessário a utilização de todo o intervalo de valores possíveis de cada parâmetro para calcular as sensibilidades dos parâmetros [Hoffman and Gardner 1983].

A Equação 1 mostra como o índice de sensibilidade é calculado pelo método de diferença percentual. A expressão  $\max \{Y(\theta)\}$  e  $\min \{Y(\theta)\}$  são, respectivamente, os valores máximo e mínimo de saída, medidos ao variar o parâmetro  $\theta$  sobre um intervalo de sua  $n$  possíveis valores de interesse. Se  $Y(\theta)$  é conhecido por variar monotonicamente, somente os valores extremos de  $\theta$  (i.e.,  $\theta_1$  e  $\theta_n$ ) precisam ser usados para calcular  $\max \{Y(\theta)\}$ ,  $\min \{Y(\theta)\}$  e como resultado  $S_\theta\{Y\}$  [de Souza Matos Júnior 2016].

$$S_\theta\{Y\} = \frac{\max \{Y(\theta)\} - \min \{Y(\theta)\}}{\max \{Y(\theta)\}} \quad (1)$$

### 3. Arquitetura de um Hospital Inteligente

A Figura 2(a) apresenta uma arquitetura de um hospital inteligente que utiliza um sistema de monitoramento de saúde baseado em IoT. As informações vitais de pacientes são coletadas por sensores conectados ao corpo. Esses dados também podem ser complementados com informações de contexto como data, hora, local, temperatura, etc. Conhecer o contexto nos permite identificar padrões incomuns e fazer inferências sobre a situação. Outros sensores e atuadores também podem ser conectados aos sistemas para transmissão de dados à equipe médica, como imagens de alta resolução.



**Figura 2. Visão Geral da Arquitetura Avaliada**

Essa arquitetura consiste em uma WSN, um *gateway*, um supervisor e dois servidores (local e remoto). A WSN é responsável por coletar os sinais biomédicos e de contexto provenientes de sensores localizados no corpo dos pacientes e no próprio ambiente. Estes dados podem ser utilizados para tratamento e diagnóstico, e são transmitidos

ao *gateway* através de protocolos de comunicação sem fio. O *gateway* deve suportar diferentes protocolos de comunicação e atua como um ponto de contato entre a WSN e o supervisor e o *switch*. O *gateway* pode receber dados de diferentes sub-redes, e deve realizar a conversão de protocolo de comunicação e fornecer outros serviços de nível superior, como agregação de dados, filtragem e assim por diante.

O supervisor pode ser um smartphone ou um tablet por onde o médico ou enfermeiro acompanha o estado dos pacientes. O *switch* é responsável por criar rotas para o local server e remote server. O local server garante que os dados sejam armazenados para pesquisas posteriores, enquanto o remote server garante a escalabilidade desses dados, além de prover mais segurança, servindo com um ponto de backup. A Figura 2(b) apresenta a arquitetura em camadas dos servidores, incluindo (*hardware, operating system, hypervisor, virtual machine, data process e storage*). Devemos destacar os componentes de software *data process e storage*, responsáveis pelo processamento e armazenamento dos dados dos sensores, respectivamente. A arquitetura em camadas apresenta a dependência dos componentes, destacando que o componente da camada superior só estará disponível caso o componente inferior esteja funcionando.

## 4. Modelo Proposto

Esta seção apresenta o modelo SPN proposto para representar o relacionamento entre os componentes das arquitetura apresentada anteriormente.

### 4.1. Modelo SPN

A Figura 3 apresenta um modelo SPN para a arquitetura de um hospital inteligente. O modelo SPN apresentado é composto por *switch* (SW), *gateway* (GA), *sensors* (SE), *supervisor* (SU), *local server* (LS) e *remote server* (RS). O *local server* e *remote server* possuem vários componentes interligados para que o sistema funcione. O *hardware* (HW), *operating system* (OS), *hypervisor* (HV), *virtual machine* (VM), *data process* (DP) e *storage* (S), fazem parte da composição do LS e RS. Cada componente possui um MTTF (tempo médio até a falha) e um MTTR (tempo médio até o reparo). As transições imediatas são utilizadas na comunicação entre os componentes que compõem o *local server* e o *remote server*. Essas transições garantem que se um componente falhar os demais componentes que dependem dele também irão falhar.

O número de sensores ativos é dado pela quantidade de tokens em SE\_U (sensores ativos). O número de tokens no lugar SE\_D (sensores desativados) representa a quantidade de sensores inativos. A alteração entre o estado de ativo e inativo para cada sensor é ocasionada pelo disparo das transições: SE\_MTTF (MTTF do sensor) e SE\_MTTR (MTTR do sensor). Consideramos que o *gateway* está funcionando quando possui um token no local GA\_U (*gateway* ativo). Consideramos que o *gateway* não está funcionando quando possui um token no local GA\_D (*gateway* desativado). As alterações de ativo/inativo são ocasionadas pelos disparos das transições: GA\_MTTF (MTTF do *gateway*) e GA\_MTTR (MTTR do *gateway*). Similar ao *gateway*, consideramos que o *switch* está funcionando quando possui um token no local SW\_U (*switch* ativo). Consideramos que o *switch* não está funcionando quando possui um token no local SW\_D (*gateway* desativado). As alterações de ativo/inativo são ocasionadas pelos disparos das transições SW\_MTTF (MTTF do *gateway*) e SW\_MTTR (MTTR do *gateway*). O Supervisor está

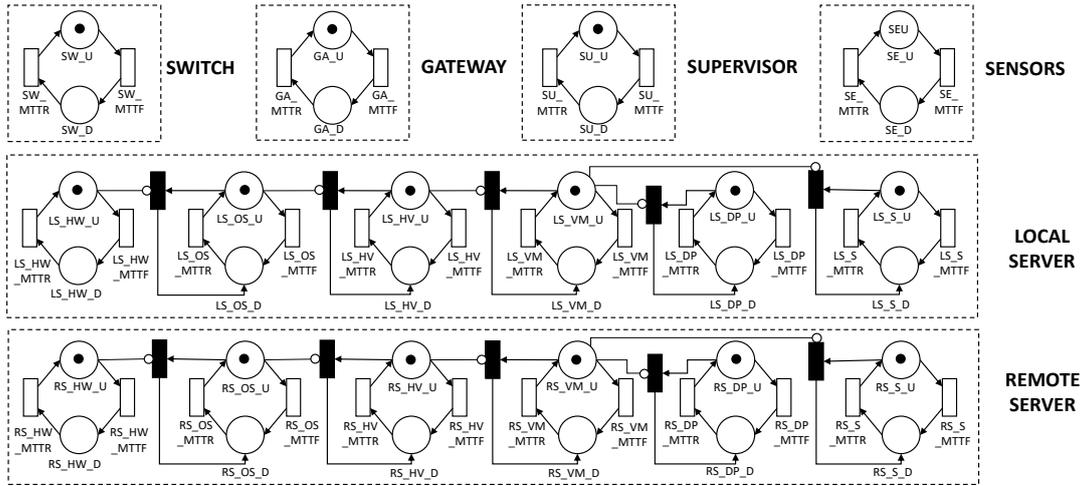


Figura 3. Modelo SPN de um Hospital Inteligente

funcionando quando possui um token no local  $SU\_U$  (*supervisor* ativo). Consideramos que o Supervisor não está funcionando quando possui um token no local  $SU\_D$  (*supervisor* desativado). As alterações de ativo/inativo são ocasionadas pelos disparos das transições  $SU\_MTTF$  (MTTF do *supervisor*) e  $SU\_MTTR$  (MTTR do *supervisor*).

O Local Server estará funcionando quando possuir tokens nos locais  $LS\_DP\_U$  (*data process* ativo) e  $LS\_S\_U$  (*storage* ativo). Consideramos que o *local server* não está funcionando quando possui um token em um dos seguintes locais:  $LS\_HW\_D$  (*hardware* desativado) ou  $LS\_OS\_D$  (*operating system* desativado) ou  $LS\_HV\_D$  (*hypervisor* desativado) ou  $LS\_VM\_D$  (*virtual machine* desativado) ou  $LS\_DP\_D$  (*data process* desativado) ou  $LS\_S\_D$  (*storage* desativado). As alterações de ativo/inativo são ocasionadas pelos disparos das transições:  $LS\_HW\_MTTF$ ,  $LS\_OS\_MTTF$ ,  $LS\_HV\_MTTF$ ,  $LS\_VM\_MTTF$ ,  $LS\_DP\_MTTF$  e  $LS\_S\_MTTF$  — para tempo médio até a falha — e  $LS\_HW\_MTTR$ ,  $LS\_OS\_MTTR$ ,  $LS\_HV\_MTTR$ ,  $LS\_VM\_MTTR$ ,  $LS\_DP\_MTTR$  e  $LS\_S\_MTTR$  — para tempo médio até o reparo.

O Remote Server estará funcionando quando possuir tokens nos locais:  $RS\_DP\_U$  (*data process* ativo) e  $RS\_S\_U$  (*storage* ativo). Consideramos que o Remote Server não está funcionando quando possui um token em um dos seguintes locais:  $RS\_HW\_D$  (*hardware* desativado) ou em  $RS\_OS\_D$  (*operating system* desativado) ou em  $RS\_HV\_D$  (*hypervisor* desativado) ou em  $RS\_VM\_D$  (*virtual machine* desativado) ou em  $RS\_DP\_D$  (*data process* desativado) ou em  $RS\_S\_D$  (*storage* desativado). As alterações de ativo/inativo são ocasionadas pelos disparos das transições:  $RS\_HW\_MTTF$ ,  $RS\_OS\_MTTF$ ,  $RS\_HV\_MTTF$ ,  $RS\_VM\_MTTF$ ,  $RS\_DP\_MTTF$  e  $RS\_S\_MTTF$  — para tempo médio até a falha — e  $RS\_HW\_MTTR$ ,  $RS\_OS\_MTTR$ ,  $RS\_HV\_MTTR$ ,  $RS\_VM\_MTTR$ ,  $RS\_DP\_MTTR$  e  $RS\_S\_MTTR$  — para tempo médio até o reparo.

A Tabela 1 apresenta condições de guarda utilizadas para o funcionamento do sistema. As condições de guarda são utilizadas nas transições e garante que elas só sejam disparadas com uma condição específica. No modelo são adotadas condições por conta da dependência de alguns componentes. Como exemplo, podemos observar que um evento de solucionar um problema em um sistema operacional só pode ocorrer se o

hardware não estiver com problema. A condição de guarda garante que o modelo conserve comportamentos que ocorram em contextos reais.

**Tabela 1. Condições de Guarda que Restringem o Disparo das Transições**

Server	Transição	Condição
Local	LS_OS_MTTR	$P\{\#LS\_HW\_U>0\}$
	LS_HV_MTTR	$P\{\#LS\_OS\_U>0\}$
	LS_VM_MTTR	$P\{\#LS\_HV\_U>0\}$
	LS_DP_MTTR	$P\{\#LS\_VM\_U>0\}$
	LS_S_MTTR	$P\{\#LS\_VM\_U>0\}$
Remote	RS_OS_MTTR	$P\{\#RS\_HW\_U>0\}$
	RS_HV_MTTR	$P\{\#RS\_OS\_U>0\}$
	RS_VM_MTTR	$P\{\#RS\_HV\_U>0\}$
	RS_DP_MTTR	$P\{\#RS\_VM\_U>0\}$
	RS_S_MTTR	$P\{\#RS\_VM\_U>0\}$

## 5. Análise Numérica

Esta seção apresenta uma análise numérica a partir do modelo proposto, incluindo uma análise de disponibilidade e uma análise de sensibilidade.

### 5.1. Definição de Cenários e Métricas

A Figura 4 apresenta os estados dos servidores em três cenários propostos (A, B, C). O cenário A retrata a métrica de disponibilidade de todo o sistema com servidor local e remoto funcionando. O cenário B retrata a métrica de disponibilidade do sistema somente com servidor local funcionando. O cenário C retrata a métrica de disponibilidade do sistema somente com o servidor remoto funcionando.

Up	Local Remote 	Local 	Remote 
Down		Remote 	Local 
	Cenário A	Cenário B	Cenário C

**Figura 4. Estados dos Servidores em Cada Cenário**

Nesse trabalho adotamos duas métricas: disponibilidade e tempo de inatividade (downtime), considerando horas por ano. A equação de disponibilidade representa uma soma de probabilidades do número de tokens de cada estado. Assim,  $P$  significa probabilidade e  $\#$  significa o número de tokens em um determinado local. A Tabela 2 apresenta as métricas utilizadas para obter a disponibilidade do sistema. O downtime ( $D$ ) pode ser obtido por  $D = (1 - A) \times 8760$ , onde  $A$  representa a disponibilidade do sistema, e 8760 a quantidade de horas no ano.

### 5.2. Resultados

Para avaliar o modelo proposto, são necessários alguns parâmetros de entrada. Os tempos de falha e reparo dos componentes são extraídos de [Araujo et al. 2019] e

**Tabela 2. Métricas Utilizadas**

Cenário	Métrica de Disponibilidade
A	$A = P\{((\#RD\_HW\_U > 0) \wedge (\#RD\_OS\_U > 0) \wedge (\#RD\_HV\_U > 0) \wedge (\#RD\_VM\_U > 0) \wedge (\#RD\_DP\_U > 0) \wedge (\#RD\_S\_U > 0)) \vee ((\#LD\_HW\_U > 0) \wedge (\#LD\_OS\_U > 0) \wedge (\#LD\_HV\_U > 0) \wedge (\#LD\_VM\_U > 0) \wedge (\#LD\_DP\_U > 0) \wedge (\#LD\_S\_U > 0))\} \wedge (\#GA\_U > 0) \wedge (\#SW\_U > 0) \wedge (\#SE\_U = SEU) \wedge (\#SUP\_UP > 0)\}$
B	$A = P\{((\#LD\_HW\_U > 0) \wedge (\#LD\_OS\_U > 0) \wedge (\#LD\_HV\_U > 0) \wedge (\#LD\_VM\_U > 0) \wedge (\#LD\_DP\_U > 0) \wedge (\#LD\_S\_U > 0)) \wedge (\#GA\_U > 0) \wedge (\#SW\_U > 0) \wedge (\#SE\_U = SEU) \wedge (\#SUP\_UP > 0)\}$
C	$A = P\{((\#RD\_HW\_U > 0) \wedge (\#RD\_OS\_U > 0) \wedge (\#RD\_HV\_U > 0) \wedge (\#RD\_VM\_U > 0) \wedge (\#RD\_S\_U > 0) \wedge (\#RD\_DP\_U > 0)) \wedge (\#GA\_U > 0) \wedge (\#SW\_U > 0) \wedge (\#SE\_U = SEU) \wedge (\#SUP\_UP > 0)\}$

[da Silva Lisboa et al. 2018], com exceção dos parâmetros do switch e do *gateway* que foram extraídos do site da Cisco, referentes ao roteador Cisco 4461<sup>2</sup> e ao switch Cisco Catalyst 1000 Series de 24-port models<sup>3</sup>. A Tabela 3 apresenta todos os valores de MTTF e MTTR utilizados neste trabalho.

**Tabela 3. Parâmetros de Entrada do Modelo**

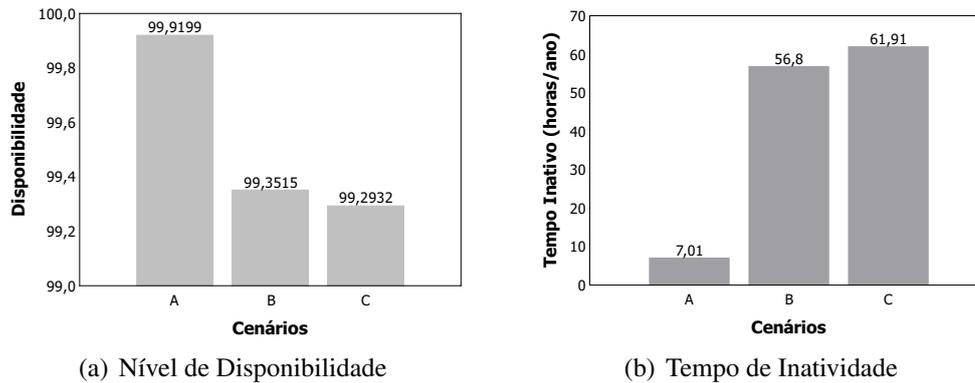
Componente	MTTF (horas)	MTTR (horas)
Switch	698220	8
Gateway	480770	8
Sensors	300000	1
Supervisor	44957	1
Hardware (LS)	4765	3
Hardware (RS)	8760	8
Operating System (LS, RS)	2800	1
Hypervisor (LS, RS)	2900	1
Virtual Machine (LS, RS)	2880	0,0958333333
Data Process (LS, RS)	700	1
Storage (LS, RS)	1440	1

Considerando os três cenários descritos anteriormente, a Figura 5 apresenta os resultados de disponibilidade e tempo de inatividade para um hospital inteligente. Os resultados foram obtidos por simulação utilizando a ferramenta Mercury [Silva et al. 2015a]. O cenário A apresenta a melhor disponibilidade (99,9199%), seguido pelo cenário B (99,3515%) e pelo cenário C (99,2932%). Isso significa que o cenário A apresenta 7,01 h/ano de inatividade, enquanto o cenário B tem 56,8 h/ano, e o cenário C, 61,91 h/ano. O cenário A apresenta a maior disponibilidade porque possui uma redundância dos servidores (local server e remote server), dessa forma, caso um dos dois servidores fique inativo o outro pode suprir as necessidades do sistema. O cenário B teve maior disponibilidade do que o C, influenciado pelo MTTR menor (3h), em relação ao cenário C (8h).

Para a análise de sensibilidade, utilizou-se os tempos de MTTF e MTTR do modelo como parâmetros, variando-os em intervalos de 10% até atingir 50% para mais e

<sup>2</sup>Disponível em: [https://www.cisco.com/c/en/us/products/collateral/routers/4000-series-integrated-services-routers-isr/data\\_sheet-c78-732542.html](https://www.cisco.com/c/en/us/products/collateral/routers/4000-series-integrated-services-routers-isr/data_sheet-c78-732542.html)

<sup>3</sup>Disponível em: <https://www.cisco.com/c/en/us/products/collateral/switches/catalyst-1000-series-switches/nb-06-cat1k-ser-switch-ds-cte-en.html?oid=otren019232>



**Figura 5. Níveis de Disponibilidade e Tempo de Inatividade em Relação ao Sistema de Monitoramento de Saúde para os Cenários Propostos**

para menos sobre o valor base. Assim, a Tabela 4, em ordem decrescente, apresenta as transições com maiores índices de sensibilidade em cada cenário, ou seja, que possuem maior impacto sobre a disponibilidade do sistema. O componente de software *data process* (DP) apresentou alto índice de sensibilidade nos três cenários, significando que a replicação desse componente é uma opção para evitar indisponibilidade do sistema.

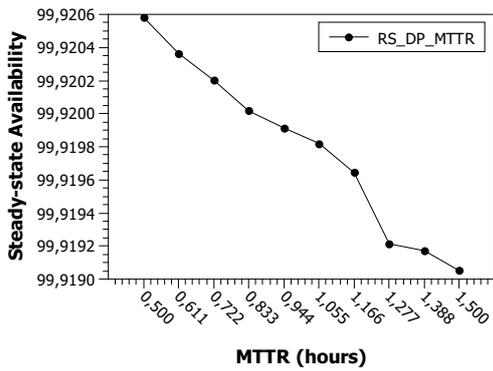
**Tabela 4. Índices de Sensibilidade**

A (Arquitetura Base)		B (Servidor Local)		C (Servidor Remoto)	
Variável	Índice	Variável	Índice	Variável	Índice
RS_DP_MTTR	$1,52 \times 10^{-5}$	LS_DP_MTTR	$2,34 \times 10^{-3}$	RS_DP_MTTR	$2,26 \times 10^{-3}$
LS_DP_MTTF	$1,51 \times 10^{-5}$	LS_HW_MTTF	$1,95 \times 10^{-3}$	RS_DP_MTTF	$1,88 \times 10^{-3}$
LS_S_MTTR	$1,11 \times 10^{-5}$	LS_DP_MTTF	$1,90 \times 10^{-3}$	RS_OS_MTTF	$1,70 \times 10^{-3}$
RS_DP_MTTF	$1,09 \times 10^{-5}$	LS_OS_MTTF	$1,70 \times 10^{-3}$	RS_S_MTTR	$1,52 \times 10^{-3}$
RS_OS_MTTF	$1,06 \times 10^{-5}$	LS_S_MTTR	$1,61 \times 10^{-3}$	RS_HV_MTTF	$1,20 \times 10^{-3}$
LS_OS_MTTF	$1,03 \times 10^{-5}$	LS_HV_MTTF	$1,20 \times 10^{-3}$	RS_HW_MTTF	$1,16 \times 10^{-3}$

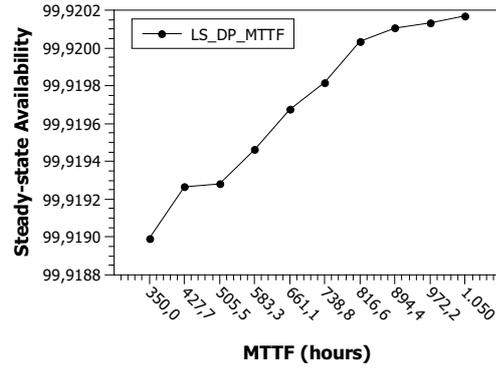
Para melhor ilustrar a influência dos parâmetros, a Figura 6 representa a variação da disponibilidade dos principais parâmetros considerando o cenário A. As Figuras 6(b), 6(d), 6(e) e 6(f) apresentam os parâmetros de MTTF, onde a disponibilidade tende a aumentar com o aumento do MTTF. As Figuras 6(a) e 6(c) mostram que o aumento do MTTR faz diminuir a disponibilidade. A variação dos parâmetros do modelo proporciona uma visão geral do comportamento do sistema. Identificamos que a disponibilidade do sistema sofre uma influência significativa das taxas de falha e reparo dos componentes *data process*, *storage* e *operating system*, à medida que seus parâmetros são variados. Um resultado interessante, considerando que todos estes componentes são software e usualmente menos custoso para replicar do que hardware.

## 6. Conclusão

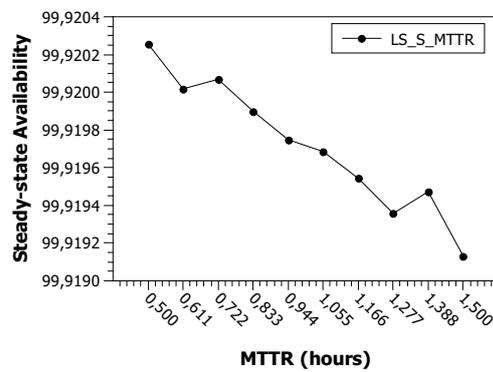
Neste artigo, propusemos um modelo SPN que representa um sistema de monitoramento de sinais vitais de pacientes em um hospital inteligente. Foram explorados três cenários onde foi possível observar o impacto que cada um apresenta na disponibilidade do sistema. Os melhores resultados foram observados no cenário A composto pela redundância



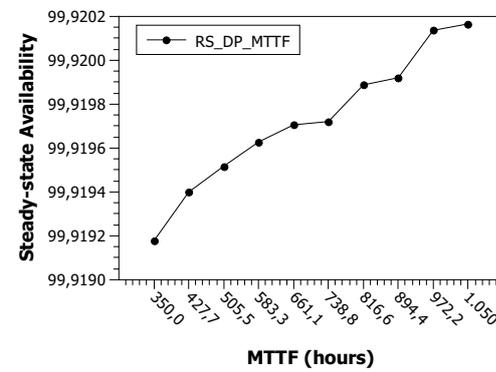
(a) RS\_DP\_MTTTR (Data Process)



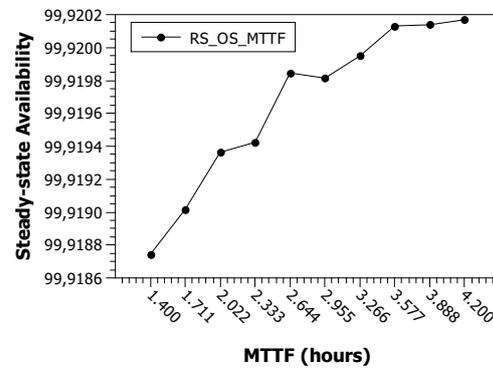
(b) LS\_DP\_MTTF (Data Process)



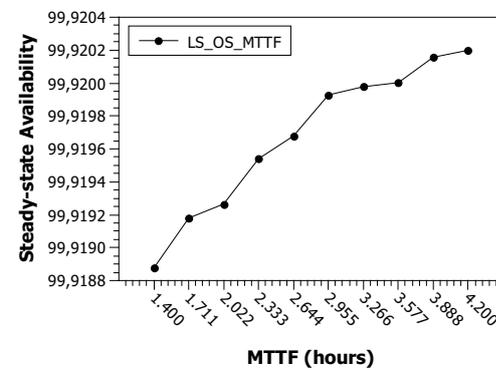
(c) LS\_S\_MTTTR (Storage)



(d) RS\_DP\_MTTF (Data Process)



(e) RS\_OS\_MTTF (Operating System)



(f) LS\_OS\_MTTF (Operating System)

**Figura 6. Resultados da Disponibilidade Variando MTRR/MTTF**

dos servidores (local e remoto). O melhor cenário — com redundância — apresentou uma disponibilidade de 99,9199%, ou seja, 7,01 h/ano de inatividade. A análise de sensibilidade foi explorada para identificar os componentes que têm maior impacto na disponibilidade do sistema como um todo. Assim, foi medido o impacto de diferentes valores de MTTF e MTRR de cada componente. A análise de sensibilidade mostra que certos componentes têm maior impacto na disponibilidade do sistema. O componente de software *data process* apresentou alto índice de sensibilidade nos três cenários, significando que tal componente deve ser replicado para evitar indisponibilidade do sistema. Outras ações podem melhorar a disponibilidade como a obtenção de componentes com melhores MTTF.

Como trabalho futuro pretende-se combinar os modelos de desempenho e disponibilidade de forma conjunta explorando questões de dependabilidade no contexto e-health.

## Referências

- Araujo, E., Dantas, J., Matos, R., Pereira, P., and Maciel, P. (2019). Dependability evaluation of an iot system: A hierarchical modelling approach. In *2019 IEEE International Conference on Systems, Man and Cybernetics (SMC)*, pages 2121–2126. IEEE.
- Araujo, J., Silva, B., Oliveira, D., and Maciel, P. (2014). Dependability evaluation of a mhealth system using a mobile cloud infrastructure. *2014 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 1348–1353.
- Campolongo, F., Tarantola, S., and Saltelli, A. (1999). Tackling quantitatively large dimensionality problems. *Computer Physics Communication*, 117(1):75–85.
- Campolongo, F., Tarantola, S., Saltelli, A., and Ratto, M. (2004). *Sensitivity analysis in practice: a guide to assessing scientific models*. John Wiley and Sons.
- Chen, L. and Ha, W. (2018). Reliability prediction and qos selection for web service composition. *IJCSE*, 16(2):202–211.
- da Silva Lisboa, M. F. F., Santos, G. L., Lynn, T., Sadok, D., Kelner, J., Endo, P. T., et al. (2018). Modeling the availability of an e-health system integrated with edge, fog and cloud infrastructures. In *2018 IEEE Symposium on Computers and Communications (ISCC)*, pages 00416–00421. IEEE.
- de Souza Matos Júnior, R. (2016). *Identification of Availability and Performance Bottlenecks in Cloud Computing Systems: An Approach Based On Hierarchical Models and Sensitivity Analysis*. PhD thesis, Federal University of Pernambuco, Center for Informatics, Graduate in Computer Science, Recife.
- Farahani, B., Firouzi, F., and Chakrabarty, K. (2020). Healthcare iot. In *Intelligent Internet of Things*, pages 515–545. Springer.
- Gomez-Sacristan, A., Rodriguez-Hernandez, M. A., and Sempere, V. (2015). Evaluation of quality of service in smart-hospital communications. *Journal of Medical Imaging and Health Informatics*, 5(8):1864–1869.
- Hoffman, F. and Gardner, R. (1983). *Evaluation of Uncertainties in Environmental Radiological Assessment Models*. Radiological Assessments.
- Kim, S.-M. (2018). A study on the design plan of ux for the smart healthcare for the aged society-focused on iot technology. *The Journal of the Korea Contents Association*, 18(11):462–474.
- Pantelopoulos, A. and Bourbakis, N. (2009). Spn-model based simulation of a wearable health monitoring system. *31st Annual International Conference of the IEEE EMBS*.
- Pianosi, F., Beven, K., Freer, J., Hall, J., Rougier, J., Stephenson, D., and Wagener, T. (2016). Sensitivity analysis of environmental models: A systematic review with practical workflow. *Environmental Modelling and Software*, 79:214–232.
- Rahmani, A. M., Gia, T. N., Negash, B., Anzanpour, A., Azimi, I., Jiang, M., and Liljeborg, P. (2018). Exploiting smart e-health gateways at the edge of healthcare internet-of-things: A fog computing approach. *Future G. C. S.*, 78:641–658.

- Rahmani, A.-M., Thanigaivelan, N. K., Gia, T. N., Granados, J., Negash, B., Liljeberg, P., and Tenhunen, H. (2015). Smart e-health gateway: Bringing intelligence to internet-of-things based ubiquitous healthcare systems. In *2015 12th Annual IEEE Consumer Communications and Networking Conference (CCNC)*, pages 826–834. IEEE.
- Rodrigues, L., Endo, P. T., and Silva, F. A. (2019). Stochastic model for evaluating smart hospitals performance. In *2019 IEEE LATINCOM*, pages 1–6. IEEE.
- Santos, G., Gomes, D., Kelner, J., Sadok, D., Silva, F., Endo, P., and Lynn, T. (2018a). The internet of things for health care: Optimizing e-health system availability in the fog and cloud. *Int. J. of Computational Science and Engineering*, pages 1–13.
- Santos, G. L., Endo, P. T., da Silva Lisboa, M. F. F., da Silva, L. G. F., Sadok, D., Kelner, J., Lynn, T., et al. (2018b). Analyzing the availability and performance of an e-health system integrated with edge, fog and cloud infrastructures. *Journal of Cloud Computing*, 7(1):16.
- Silva, B., Matos, R., Callou, G., Figueiredo, J., Oliveira, D., Ferreira, J., Dantas, J., Lobo, A., Alves, V., and Maciel, P. (2015a). Mercury: An integrated environment for performance and dependability evaluation of general systems. In *Proceedings of Industrial Track at 45th Dependable Systems and Networks Conference, DSN*.
- Silva, F. A., Kosta, S., Rodrigues, M., Oliveira, D., Maciel, T., Mei, A., and Maciel, P. (2017). Mobile cloud performance evaluation using stochastic models. *IEEE Transactions on Mobile Computing*, 17(5):1134–1147.
- Silva, F. A., Rodrigues, M., Maciel, P., Kosta, S., and Mei, A. (2015b). Planning mobile cloud infrastructures using stochastic petri nets and graphic processing units. In *2015 IEEE 7th International Conference on Cloud Computing Technology and Science (CloudCom)*, pages 471–474. IEEE.
- Tigre, M., Santos, G., Lynn, T., Sadok, D., Kelner, J., and Endo, P. (2018). Modeling the availability of an e-health system integrated with edge, fog and cloud infrastructures. *IEEE ISCC*, pages 416–421.
- Wong, J., Goh, Q. Y., Tan, Z., Lie, S. A., Tay, Y. C., Ng, S. Y., and Soh, C. R. (2020). Preparing for a covid-19 pandemic: a review of operating room outbreak response measures in a large tertiary hospital in singapore. *Canadian Journal of Anesthesia/Journal canadien d’anesthésie*, pages 1–14.
- Xiaoli, W., Guangju, C., Quiang, Z., and Zhongping, G. (2007). Reduction of stochastic petri nets for reliability analysis. *The Eighth International Conference on Electronic Measurement and Instruments*.
- Zhang, H., Li, J., Wen, B., Xun, Y., and Liu, J. (2018). Connecting intelligent things in smart hospitals using nb-iot. *IEEE Internet of Things Journal*, 5(3):1550–1560.