ConBRepro
X CONGRESSO BRASILEIRO DE ENGENHARIA DE PRODUÇÃO
EVENTO ON-LINE
02 a 04 de dezembro 2020

Sistemas Tolerantes a Falhas para Servidores e Redes de Computadores Aplicáveis aos Sistemas de Informação Hospitalares

Eliéser Paiva de Sousa Júnior[1], Rafael Lemos Pereira[2], Adan Lucio Pereira[3]

[1,2,3] Faculdade Brasileira - Multivix

(elieserpaivajr@hotmail.com[1], rafa.rois@hotmail.com[2], adanlucio@gmail.com[3])

Resumo: A disponibilidade de dados e informações no ambiente hospitalar tem sido cada vez mais importante. Com os avanços tecnológicos, os documentos físicos estão cada vez mais em desuso. Exames, parecer médicos, prontuários, entre outros registros são inseridos em banco de dados dos sistemas por diversos terminais espalhados em uma unidade. A comunicação entre todos os dispositivos e a alta disponibilidade de informações em tempo integral é imprescindível para o bom funcionamento dos processos e procedimentos médicos. A falha de um equipamento ou interrupção da comunicação entre eles poder refletir diretamente na vida de um paciente. Diante deste cenário, este trabalho apresenta soluções para estabelecer um sistema tolerante a falhas, proporcionando diversas camadas de proteção através de redundância de hardwares, softwares e dados, com o intuito principal de aumentar a confiabilidade de toda estrutura do sistema de informação.

Palavras-chave: Alta Disponibilidade, Tolerante a Falha, Confiabilidade, Sistema de Informação.

Fault Tolerant Systems for Servers and Computer Networks Applicable to Hospital Information Systems

Abstract: The availability of data and information in the hospital environment has been increasingly important. With technological advances, physical documents are increasingly in disuse. Examinations, medical opinion, medical records, among other records are inserted in the systems' database by several terminals spread over a unit. Communication between all devices and the high availability of information on a full-time basis is essential for the proper functioning of medical processes and procedures. The failure of an equipment or interruption of communication between them can directly reflect on a patient's life. Against

this scenario, this work presents solutions to set a fault-tolerant system, establishing several layers of protection through redundancy of hardware, software and data, with the main purpose of increasing the reliability of the entire information system structure.

Keywords: High Availability, Fault Tolerant, Reliability, Information system.

1. Introdução

Na atualidade, a computação está presente em quase todos os empreendimentos (SILVA *et al*, 2019). Segundo pesquisa realizada pelo SEBRAE (2015), cerca de 76% dos empresários já utilizavam computadores em seus negócios. Nesse aspecto, cada vez mais as empresas são dependentes de sistemas, acesso à internet e outros recursos interligados à rede (XAVIER; CARVALHO, 2014). O uso e a dependência dos recursos tecnológicos, que atinge desde as empresas até um usuário doméstico, estão cada vez maiores.

Todos esses avanços produzem um grande volume de dados e informações que dependem de *softwares* e *hardware* para manter os sistemas em funcionamento. Estas ferramentas convergem para uma dependência cada vez maior da disponibilidade e consequentemente o bom funcionamento dos computadores.

O Sistema de Informação Hospitalar (SIH) é um sistema computacional imprescindível ao ambiente hospitalar, uma vez que, estes assistem o diagnóstico médico, integram informações, auxiliam a melhoria no atendimento de um serviço de necessidade básica, bem como a gestão da saúde por meio da análise dos custos e benefícios. Porém, é necessário garantir a integridade das informações armazenadas, eficácia e impactos de sua aplicação; com objetivo de prevenir consequências penosas, como indução ao erro médico ou processos judiciais (PEREIRA et al, 2012).

Contudo, todos os equipamentos e *softwares* estão sujeitos a falhas físicas ou humanas e que podem ser causados por fadiga de componentes, variações ambientais, problemas de implementação, interferências eletromagnéticas, entre outras (MAGALHAES; PINHEIRO, 2008). Corroborando, uma pesquisa realizada pela *Enterprise Strategy Group* – ESG (2018) em parceria com a Dell, apenas 6% das empresas do mundo estão na faixa considerada totalmente transformadas na curva de maturidade de TI. Esses fatos mostram o quanto é necessário evoluir para que os recursos computacionais, que tanto auxiliam os empreendimentos, não sejam também causadores de prejuízos.

Essa possibilidade de falha pode ser evitada aplicando ferramentas de redundância de informação, onde softwares e hardwares garantem a preservação dos dados e estabilidade do sistema (KOREN; KRISHNA, 2010). Inserido nesse contexto, o presente trabalho visa analisar os sistemas tolerantes a falhas para servidores e redes de computadores aplicáveis aos ambientes hospitalares. Para isso serão mapeados os sistemas tolerantes a falhas utilizados na literatura. Será proposto soluções de redundância em *hardware* e *softwares* para servidores e apresentar recursos de balanceamento de *links* de internet e redundância de rede

2. Sistemas Tolerantes a Falha

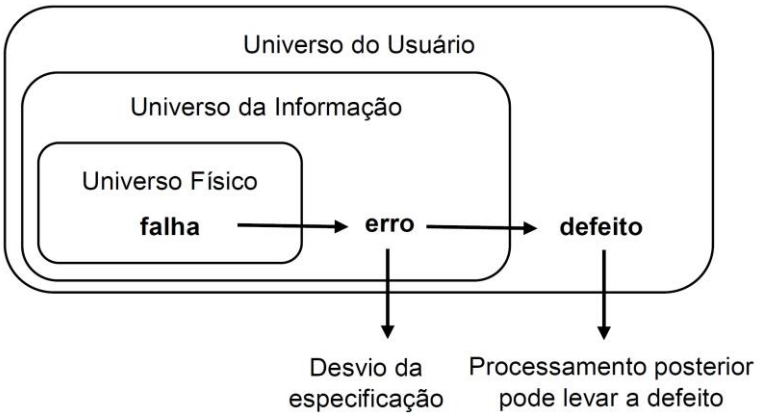
A palavra falha constitui uma ausência de perfeição, incorreção ou até mesmo falha de desenvolvimento. Nesse aspecto, os componentes físicos de um computador que lidam com interferências externas e envelhecem estão sujeitos a falhar em algum momento.

Os softwares que fazem a interação entre o usuário e o hardware possuem alta complexibilidade e também estão sujeitos a falhas. Problemas de implementação ou de especificação, componentes defeituosos e fadigas de componentes físicos estão na lista dos principais motivos das falhas. Outra abordagem para a definição desse conceito pode

ser definida quando um sistema se desvia da especificação que foi originalmente projetada (WEBER, 2002).

Essa relação de falha em um meio físico até ser observada como um defeito por um usuário é relatada por Magalhães e Pinheiro (2008), conforme ilustrado na Figura 1.

Figura 1 - Relação entre falha, erro e defeito.



Fonte: Adaptado de Magalhaes e Pinheiro (2008).

Nesse contexto, as falhas podem ter origem no (universo físico), como um chip de memória que apresenta falha em um de seus bits, ocasionando uma interpretação errada no dado armazenado em uma estrutura (universo da informação), interrompendo o embarque de passageiros de um vôo (universo do usuário) por considerar que o vôo estava lotado. Esse tipo de informação errada pode ser contornado a partir de redundâncias na estrutura do sistema. Neste aspecto, o Quadro 1 apresenta vários conceitos que foram a base dos sistemas tolerantes a falha.

Quadro 1 – Conceitos de sistemas tolerantes a falha.

Sistemas Tolerantes a Falha	Referência
Sistemas projetados com elementos adicionais e algoritmos especiais que garantem um correto funcionamento do sistema mesmo em caso de falha de algum componente. Esse modelo é muito usado em ambientes que necessitam de alta disponibilidade e confiabilidade.	(WEBER, 2003)
Sistemas capazes de computar corretamente independentemente da existência de erros. Em geral, qualquer sistema que contenha funções e componentes redundantes possui algumas propriedades de tolerância a falha	(SHOOMAN, 2003).
Sistema redundante pode ocultar a presença de falhas, pois possuem mecanismos dedicados a contornar alguns problemas. O usuário espera que o sistema esteja sempre à disposição quando requisitado. Sendo assim, é necessário aplicar estratégias para que uma interrupção indesejada não ocorra tão facilmente.	(COSTA, 2009)
As estratégias utilizadas para sistemas tolerantes a falhas não se restringem apenas a alguns pontos críticos como armazenamento de dados, *layout* da rede ou fornecimento de energia, mas, toda a estrutura física dos equipamentos e a infraestrutura do ambiente são levados em consideração para garantir uma alta disponibilidade ao usuário	(FERREIRA, et al., 2005)

Fonte: Elaborado pelo Autor.

2.1 Tipos de redundância

Umas das principais estratégias utilizadas na tolerância de falhas é a redundância. Quatro segmentos de redundância são descritos por Weber (2003) O primeiro deles trata a Redundância de informação, onde as informações (bits extras) são enviadas junto aos dados para localização de erro ou ocultação de falhas. Em seguida tem-se a redundância temporal, onde um reprocessamento é realizado para a verificação de possíveis erros ou

falhas. Nessa abordagem possui um aumento no tempo de computação, porém, não depende de custo de hardware.

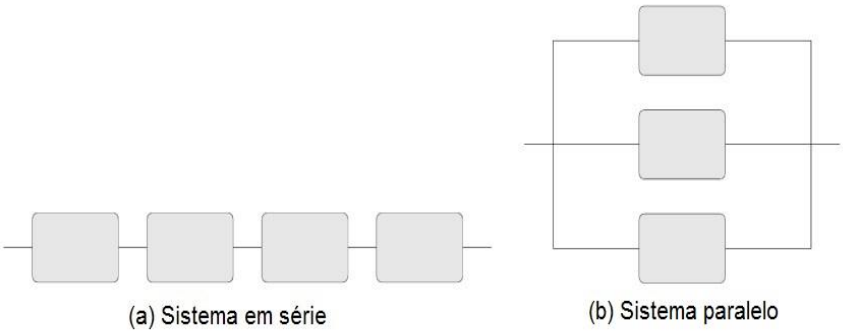
O Terceiro tipo consiste na redundância de hardware, onde dois ou mais componentes são utilizados em um mesmo sistema de forma paralela ou serial. E por fim pode ser apresentada a redundância de software em que os algoritmos, que são desenvolvidos com ferramentas de bloco de recuperação, verificação de consistência e programação n-versões, com o objetivo de detectar ou mascarar falhas.

O primeiro tipo de redundância citado é aplicado na fase de fabricação do componente, e o segundo tipo demanda tempo de computação, logo, tratam de uma abordagem distinta da proposta deste trabalho, razão pela qual eles não serão detalhados neste estudo.

2.1.1 Hardwares tolerantes a falha

Os mais básicos tipos de construção de sistemas são aqueles organizados em paralelo e serial. Esses sistemas básicos são apresentados na Figura 2.

Figura 2 - Sistemas serial e paralelo.



Fonte: (KOREN; KRISHNA, 2010).

Na Figura 2a é observado um diagrama onde os blocos, exceto o primeiro, são realimentados pelo bloco anterior e não pela entrada inicial do sistema. Para esse tipo de sistema ser livre de falhas, é preciso que os quatro módulos sejam íntegros. Se qualquer um deles falhar, todo o sistema estará comprometido, não sendo apropriado para um tipo de sistema crítico que busca ser tolerantes a falha. Na Figura 2b a organização foi realizada de forma que todos os módulos estão eletricamente conectados na entrada e saída do sistema. Dessa forma, para ocorrer uma falha é necessário que todos os módulos falhem (KOREN; KRISHNA, 2010).

Essa modelagem em paralelo reduz drasticamente a possibilidade de falha do sistema por possuir um *layout* redundante, ou seja, vários componentes em conjunto habilitados a realizarem a mesma tarefa. Um hardware, que é desenvolvido neste padrão, apresenta uma abordagem mais onerosa em relação ao custo e até um tamanho físico maior em comparação ao sistema tradicional, por possuir diversos componentes que são duplicados. Esses tipos de dispositivos são utilizados em ambientes em que a paralisação ou interrupção de um serviço possuem consequências graves, justificando o investimento econômico e o espaço físico que abrigará o componente.

Um exemplo de equipamento que é desenvolvido utilizando esse modelo de paralelismo, com o objetivo de se obter a redundância de hardware e tolerância a falha, é a linha de servidores em torre PowerEdge T340 da DELL (2020), que possui configurações que permitem a utilização duas fontes de alimentação e duas placas de redes integradas. Esses componentes duplicados mantêm o funcionamento do equipamento mesmo em caso de falha de um deles.

Outra abordagem para se obter a tolerância a falha a nível de hardware é relatada por Koren e Krishna (2010), através da utilização do método chamado *watchdog* (cão guarda ou processador de vigilância), que é aplicado ao processador. Esse tipo de arquitetura insere um módulo que monitora a atividade do processador, executando a detecção de erros através do monitoramento do barramento que interliga o processador à memória.

Essa técnica verifica os fluxos de controle para checar se as execuções dos blocos de códigos estão corretas e na ordem correta. Contudo, vale destacar, que o *watchdog* apenas detecta uma parte dos erros, pois alguns erros não alteram o fluxo normal do programa e nem o seu caminho (KOREN; KRISHNA, 2010).

O *watchdog* possui uma versão chamada de *watchdog timer*. Para LAMBERSON (2013), este dispositivo é amplamente utilizado em dispositivos embarcados, além dos processadores. Seu princípio é desempenhar a função de um contador ou um relógio que monitora o sistema. Caso seja detectado alguma falha ou erro do programa, o dispositivo atua restaurando a estabilidade do sistema. Como sua atuação acontece de forma automática, ele responde com muito mais eficiência, em comparação à atuação humana diante de um problema, tornando uma excelente ferramenta ao combate de falhas transitórias.

Por fim, o armazenamento dos dados também possui recursos tolerantes a falhas a nível de hardware. O RAID *(Redundant Array of Independent Disks* - Conjunto Redundante de Discos Independentes) pode ser implementado por hardware através do um controlador especializado. Esse controlador faz o gerenciamento e a ponte entre os conjuntos de discos do sistema e o RAID (SOMASUNDARAM *et al*., 2011).

As técnicas disponibilizadas pelo modelo RAID descritas por Somasundaram *et al*. (2011) são:

- ***Striping*:** Essa configuração possui a função específica de melhorar o desempenho de gravação e leitura das informações, uma vez que os dados armazenados são divididos em pequenos pedaços que são espalhados pelos discos.
- **Espelhamento:** Essa técnica possui um *layout* que armazena as informações em dois discos diferentes em forma simultânea. Cada disco é totalmente espelhado em outro disco, possuindo exatamente os mesmos dados. Se um dos discos falharem, a informação não é perdida, pois o outro disco assume de forma independente enquanto o disco defeituoso não é substituído.
- **Paridade:** É um método que apresenta uma dinâmica tolerante a falhas, porém, sem o custo do espelhamento total do disco. A paridade funciona através da divisão dos discos em blocos e organizados em linhas. Nesse modelo, o conjunto de discos possui um disco dedicado a realizar a paridade dos demais. Em caso de falha, é possível obter o valor perdido através da subtração do valor da paridade pela soma dos outros elementos daquela linha de blocos.

Essas técnicas possuem finalidades diferentes e devem ser aplicadas com o objetivo de se obter desempenho, tolerância a falha ou ambos. Neste aspecto, Somasundaram *et al*. (2011) destaca ainda os diversos níveis de RAID, sendo eles: RAID 0, RAID 1, RAID 3, aninhado, entre outros.

O RAID 0 utiliza a técnica *striping* com o objetivo de melhorar o desempenho, porém, não é tolerante a falhas. O RAID 1 realiza o espelhamento dos discos, permitindo a recuperação dos dados em uma operação contínua. O RAID 3 fraciona as informações entre os discos, melhorando o desempenho, e possui um disco dedicado a paridade para recuperação de dados em caso de falha. O nível aninhado é a combinação de dois níveis, como, por

exemplo, o RAID 1 e RAID 0, tornando o RAID tolerante a falhas e com aumento no desempenho.

Em resumo, o Quadro 2 demonstra uma análise comparativa entre os hardwares tolerantes a falhas, através de uma matriz SWOT. Esse tipo de comparação destaca alguns pontos fundamentais, auxiliando a compreensão e distinção da técnica ou conjunto de técnicas a ser adotada.

Quadro 2 - Análise SWOT dos hardwares tolerantes a falhas.

	S Forças	W Fraquezas	O Oportunidades	T Ameaças
Sistemas Paralelo	Proporciona alta confiabilidade e manutenção elementar	Duplicidade total do sistema ou subsistema e custo elevado	Fortemente recomendados para setores que demanda alta disponibilidade	Compatibilidade do software com o hardware
Watchdog e watchdog timer	Implementado no próprio equipamento	Pouca possibilidades de configuração	Compatível com diversas aplicações	Trata apenas os cenários implementados pelo fabricante.
RAID	Possibilita alta confiabilidade	Custos mais elevados para implantação	Compatível e escalável a pequenos e grandes sistema	Popularização dos sistemas de armazenamento em nuvens em tempo real

Fonte: Elaborado pelo Autor.

Conforme descrito no Quadro 2, nenhuma das técnicas contém apenas pontos positivos, pois cada abordagem também possui suas fraquezas. Neste sentido, é necessário realizar uma junção entre as técnicas para contemplar o sistema como um todo, envolvendo as áreas de alimentação elétrica, placas de comunicação, processamento e armazenamento.

2.1.2 Softwares tolerantes a falhas

Os softwares se correlacionam ao conceito que não possuem componentes materiais, logo, não sofrem desgastes ou envelhecimento de componentes com o passar do tempo, assim como acontece com os hardwares.

Por se tratar de um tema que é totalmente conceitual, qualquer tipo de erro é sempre causado por falha de projeto. As falhas de projetos estão relacionadas aos seguintes aspectos: qualificação dos responsáveis pelo desenvolvimento do sistema; aplicação de métodos de especificação; testes e certificação do sistema. Para se obter a confiabilidade do sistema, é necessário prever tais situações que se relacionam com essas incorreções do projeto e tomar medidas preventivas reduzindo ao máximo a possibilidade de falhas (GUERRA, 2004).

Assim como em hardware, os softwares também apresentam falhas, erros e defeitos. Para Koscianski e Soares (2007), esses conceitos podem ser definidos como:

- Quando há uma imperfeição no programa, causado geralmente por uma implementação incorreta, chama-se de **defeito**. Este faz parte do programa e pode ser recorrente por se tratar um “vício oculto” que o software apresenta. Defeito ainda por ser relacionado a situação onde o programa não funciona como o esperado.
- O **erro** faz correlação a ocorrência de uma inconsistência que é provocada por partes de um código que estão devidamente implementadas, porém, podem apresentar problemas no resultado de determinadas operações e combinação de alguns valores em específico.
- A **falha** é a consequência de um defeito. Como os programas podem ser muito extensos e conter diversas estruturas e trechos de códigos, então, é possível que um defeito nunca seja descoberto e, por consequência, o sistema não apresentará

falha, apesar dele existir. Falhas também podem ser causadas por fatores externos, como uma inconsistência no banco de dados causado por um outro programa.

Dessa forma, a tolerância a falha para softwares pode ser feita utilizando a redundância. Contudo, apenas replicar um código ou utilizar o paralelismo não contorna o problema, pois em algoritmos idênticos, se tem problemas idênticos.

Lyu (2007) afirma que existem dois grupos de técnicas para alcançar a tolerância a falhas, sendo eles: Softwares de versão única e multi-versões. O primeiro, compreende técnicas de manuseio de exceção, pontos de verificação, modulação do programa, detecção de erros e reinicialização. O segundo, é definido pela elaboração de diversas versões elaboradas por diferentes técnicas até se obter a melhor versão que será integrada ao programa.

Weber (2003) cita duas técnicas do grupo multi-versões que intensificam a confiabilidade do sistema. As técnicas são:

- **Programação em n-versões:** Essa estratégia consiste em elaborar o sistema em diversas versões. As equipes de trabalho elaboram estratégias diferentes para um mesmo cenário. Cada abordagem utiliza uma fonte de pesquisa e uma forma de criar as soluções para cada etapa. Após a conclusão da etapa, as diversas versões são testadas e classificadas para alcançar aquela que obtém melhores resultados.
- **Blocos de recuperação:** Este método é semelhante ao anterior, porém, as versões seguintes só são desenvolvidas após a reprovação da primeira. Uma versão primária é desenvolvida e é submetida a um teste de aceitação. Se essa versão não passar no teste, uma segunda versão é desenvolvida e submetida ao teste. Diversas versões são criadas até que umas delas passe no teste de aceitação.

Para o grupo dos softwares tolerantes a falhas de versão única, Dubrova (2013) afirma que esse modelo possui estrutura e ações que identificam falhas e impossibilitam a sua replicação pelo restante do sistema. Esse método é composto por três técnicas, sendo elas:

- **Detecção de falhas:** Essa técnica consiste em submeter o programa a uma série de testes de aceitação. Esses testes compreendem verificações de tempo de execução, codificação, estruturas do sistema e razoabilidade. Essas investidas contra o programa identifica as possíveis falhas do sistema.
- **Contenção de falhas:** As falhas identificadas na técnica anterior são contidas através da alteração da estrutura do sistema, que passa a possuir restrições que determinam as ações que são permitidas. A contenção é feita utilizando outras quatro técnicas: modularização (subdividindo o sistema em módulo), particionamento (cria a dimensão horizontal e vertical), fechamento do sistema (nenhuma ação é permitida, exceto as explicitamente autorizadas) e ações atômicas (interação direta entre componentes).
- **Recuperação de falhas:** Consiste na inserção de pontos de verificação do sistema, onde são realizadas marcações e, em caso de falha nas instruções seguintes, o sistema retorna ao último ponto de verificação antes da ocorrência da falha e retoma o processamento.

Essas técnicas são utilizadas para minimizar a possibilidade de falhas, mas não garante na totalidade a inexistência delas. Diversos motivos podem conduzir o sistema a apresentar um resultado diferente daquele que é esperado. Erros de implementação, utilização inadequada por parte do usuário final, inconsistência no banco de dados causado por outros programas, são exemplos de eventos que podem desencadear uma condição diferente daquela que o sistema foi projetado para tratar. Todavia, utilizando essas técnicas de forma

adequada, pode-se construir um sistema confiável e seguro que pode ser considerado tolerante a falhas.

Em síntese, o Quadro 3 demonstra uma análise comparativa entre as técnicas de softwares tolerantes a falhas.

Quadro 3 - Análise SWOT das técnicas para softwares tolerantes a falhas.

	S Forças	W Fraquezas	O Oportunidades	T Ameaças
Programa em n-versões	Inúmeras metodologias são aplicadas ao mesmo software	Custo elevado de desenvolvimento e processo complexo de sincronização de versão	Adaptabilidade a qualquer tipo de cenário	Complexidade geral do sistema
Blocos de recuperação	Modelo de desenvolvimento facilita a identificação de falhas	O desenvolvimento dos blocos seguintes dependem da aprovação do anterior	Cada bloco atende a requisitos bem definidos, flexibilizando sua implementação.	Metodologia de desenvolvimento pode gerar atrasos na entrega final do projeto
Versão única	Implementação simples e flexível	Sujeito a falha que não são recuperáveis pelos pontos de recuperação	Compatível e aplicável a qualquer tipo de cenário.	Metodologia singular de desenvolvimento pode ser menos eficiente para contornar falhas

Fonte: Elaborado pelo Autor.

Por fim, como apresentado no Quadro 3, cada estratégia possui pontos positivos e negativos. Dessa forma, cabe a organização especificar junto ao desenvolvedor ou representante do sistema qual método é mais eficiente para uma aplicação específica, a fim de se obter um software que seja confiável, seguro e tolerante a falhas.

3. Redes de Computadores Tolerantes a Falhas

3.1 Conceito geral de rede de computadores

Uma rede de computadores é a interconexão de vários equipamentos, como um computador, celular, impressora, entre outros. O aparelho responsável por realizar essa interligação entre eles pode ser definido como dispositivo de conexão, tal como um roteador, *switch* ou outro equivalente.

Quando a conexão acontece, o usuário passa a fazer parte de uma rede chamada LAN (*Local Area Network* – Rede de Área Local), que em geral é uma rede privada e interliga alguns *hosts* em um escritório, por exemplo, ou uma rede WAN (*Wide Area Network* – Rede de Longa Distância), que possui uma extensão geográfica muito maior, podendo abranger até o mundo todo, como a internet (FOROUZAN e MOSHARRAF, 2013).

Esse cenário de múltiplas conexões e dimensões globais requer um sistema universal e padronizado que possibilite a comunicação entre os diversos tipos de equipamentos. Nesse contexto, foram criados o protocolo TCP/IP (*Transmission Control Protocol/Internet Protocol* - Protocolo de Controle de Transmissão/Protocolo de Internet), que padroniza uma forma de conexão entre os diferentes equipamentos, e o modelo OSI (*Open System Interconnection* – Sistema Aberto de Interconexão), que define a arquitetura dos protocolos, entre os anos 1970 e 1980 (MORAIS, 2014).

Por consequência dessa universalidade, esses conceitos estão presentes nas redes atuais e permitem a comunicação entre os mais diversos tipos de equipamentos, o que proporciona uma diversidade de recursos e possibilidades de interação.

3.2 Topologia de redes

A topologia da rede está relacionada com a forma com que os dispositivos são organizados. A topologia comumente utilizada é a estrela, que consiste em um equipamento central que interconecta todos os demais dispositivos. Esse tipo de rede permite a inclusão ou remoção

de computadores no sistema sem nenhum tipo de interferência aos demais dispositivos (MORAIS, 2010).

Para Morais (2010), essa topologia é mais tolerante a falhas, pois permite a continuidade da rede através da possibilidade de agregar outros concentradores em um ambiente já existente, permitindo a inserção de novos setores sem a necessidade de paralização do sistema.

Este modelo centralizado permite um gerenciamento mais simplificado, pois todas as estações são conectadas aos concentradores. Outra vantagem dessa abordagem é que o equipamento central dispõe de soluções e recursos para todos os componentes da rede, dispensando configurações individualizadas, o que o torna o padrão eficiente.

3.3 Balanceamento de cargas

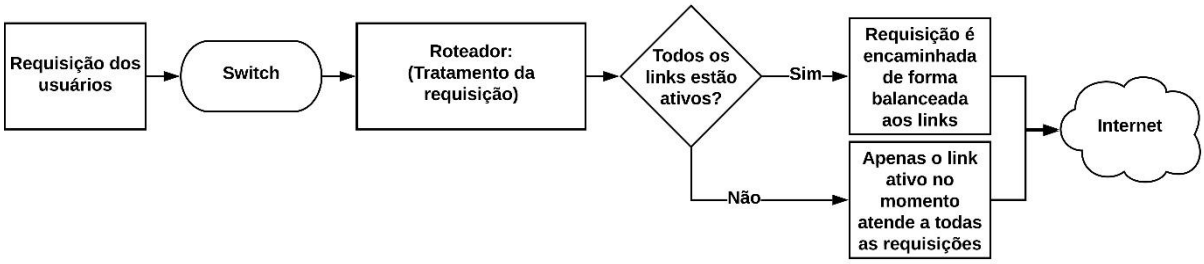
O balanceamento de carga é uma técnica capaz de reencaminhar requisições por caminhos diferentes. Esse recurso ameniza problemas de sobrecarga e interrupção de sistema, pois distribui os pedidos realizados a um servidor por mais de uma rota, de tal forma que os nós e enlaces da rede não fiquem sobrecarregados. Essa abordagem, além de proporcionar uma melhor performance de toda a rede, estabelece um serviço tolerante a falhas (OLIVEIRA; SALLES, 2016).

Desta forma, essa tecnologia permite a agregação de dois ou mais links que podem funcionar de maneira conjunta, aumentando a banda disponível, ou de forma redundantes, onde na falha de um deles o outro assume de forma independente atendendo a todas as requisições (DEJANO; BELLEZI, 2014).

Esse mecanismo pode ser realizado na camada 2 - enlace do modelo OSI, através da comutação dos pacotes que trafegam pela rede. Para os usuários, existe apenas um link virtual ativo e a negociação de pacotes acontece de forma transparente, onde o utilizador não define por qual rota a requisição irá trafegar e nem mesmo consegue perceber a queda de um dos links (FILIPPETTI, 2014).

Em resumo, o fluxograma apresentado na Figura 3 apresenta a trajetória percorrida por uma requisição através desse sistema redundante que possui dois links.

Figura 3 – Trajetória de uma requisição em um sistema balanceado.



Fonte: Elaborado pelo Autor.

Como observado na Figura 3, esse tipo de sistema demonstra um ambiente de alta disponibilidade e confiabilidade em razão da redundância de links que operaram de forma automatizada e gerenciada por um roteador, condição a qual estabelece uma rede de computadores tolerante a falhas.

4. Discussão Sobre o Sistema de Informação Hospitalar

Os Sistemas de Informação em Saúde (SIS) tem como objetivo processar, armazenar, coletar e disseminar dados que ofereça suporte ao processo decisório em saúde (CAVALCANTE et al, 2011). Isso proporciona o aperfeiçoamento das práticas desenvolvidas, além de axilar o processo de gestão (MARIN, 2010).

Os SIS devem fornecer informações relevantes, potencializar a comunicação e promover a segurança necessária no ambiente organizacional. Estas características podem auxiliar os profissionais de saúde no planejamento, bem como na tomada de decisões relacionadas à gerência e a assistência aos pacientes (MARIN, 2010). Por meio dos softwares ocorre a disponibilização das informações através de relatórios, tabelas e gráficos. Alguns softwares são capazes de oferecer conexão entre dados, o que proporciona agilidade no processo de análise e como consequência no processo decisório (CAVALCANTE et al, 2012).

No ambiente hospitalar, o meio utilizado para trocas de informação na gestão é o Sistema de Informação Hospitalar (SIH), que tem como finalidade possibilitar a comunicação dos indivíduos dentro da organização, além de oferecer contribuições para tomada de decisão. Visto que o hospital é um ambiente complexo, grande produtor de dados e informações, o SIH objetiva contemplar a necessidade de veracidade, agilidade dos dados, além de otimizar o fluxo da informação, a eficiência e a integração dos processos e consequentemente, por meio do planejamento, aprimorar a qualidade da assistência em saúde (MARIN, 2010).

O SIH deve se estabelecer como um sistema especializado e independente, capaz de integrar as informações pertinentes sobre a assistência prestada ao paciente, assim como facilitar o desempenho das atividades planejadas e o alcance das metas pretendidas. Em suma, os SIH são utilizados para dar suporte ao planejamento das ações em saúde, buscando a eficiência e a melhoria da qualidade da assistência ao paciente (GUTIERREZ, 2011).

Desta forma, a utilização de SIS é o recurso para a gestão estratégica em saúde. É o suporte para a organização administrativa e técnicas, a coleta de dados, o armazenamento, o processamento das informações dos pacientes, o auxílio ao diagnóstico, a prescrição dos medicamentos e cuidados adequados a cada situação em que o paciente estiver envolvido (CAVALCANTE et al, 2011).

5. Considerações Finais

O modelo de alta disponibilidade gerado por um sistema tolerantes a falha é essencial em ambientes hospitalares, em virtude que uma falha de um equipamento ou software pode influenciar diretamente na vida de um paciente. É necessário garantir a integridade das informações armazenadas, eficácia e impactos de sua aplicação. Uma das possibilidades para isso é utilizando as redundâncias de rede, hardware e software apresentadas neste artigo. Entretanto, quando se analisa a realidade percebe-se que a área de infraestrutura dos equipamentos de tecnologia da informação ainda carece incentivo e regulamentação.

Em síntese, a utilização de sistemas tolerantes a falha, além de tratar os riscos provenientes do modelo tradicional, esses novos sistemas devem garantir a segurança de uma nova infraestrutura, que apesar da necessidade de investimentos tais valores são mínimos quando comparados com o grau de importância das informações armazenadas e processadas em um ambiente hospitalar. Dessa forma, alinhando os recursos computacionais a segurança dos ambientes onde são aplicados é possível visualizar uma grande oportunidade de sinergia entre as áreas da saúde e tecnologia, além de fomentar a uma infraestrutura tecnológica segura aplicada a saúde.

Referências

CAVALCANTE, R. B.; SILVA, P. C.; FERREIRA, M .N. **Sistema de informação em saúde: possibilidades e desafios**. Rev Enferm UFSM. 2011;1(2):290-9.

CAVALCANTE, R. B; CUNHA, S. G. S; BERNARDES, M. F. V. G.; GONTIJO, T. L.; GUIMARÃES, E. A. A.; OLIVEIRA, V. C. **Sistema de Informação Hospitalar: utilização no processo decisório**. J. Health Inform. *2012 Julho-Setembro; 4(3): 73-9*

COSTA, Hebert Luiz Amaral et al. **Alta disponibilidade e balanceamento de carga para a melhoria de sistemas computacionais críticos usando software livre**: um estudo de caso. 2009. Disponível em: <https://www.locus.ufv.br/handle/123456789/ 2594>. Acesso em 11 mar. 2020.

DEJANO, Fernando Rezende; BELLEZI, Marcos Augusto. Alta disponibilidade: Balanceamento de carga e tolerância à falhas utilizando roteamento, firewall e qos avançado na plataforma linux. **Tecnologia, Infraestrutura e Software,** São Carlos, v3, n. 1 p.45-57, jan./abr. 2014.

DELL, **Servidor em torre PowerEdge T340**. 2020. Disponível em: < https://www.dell.com/pt-br/work/shop/productdetailstxn/poweredge-t340>. Acesso em 04 mai. 2020.

DUBROVA, Elena. **Fault-tolerant design**. New York: Springer, 2013.

ESG. Enterprise Strategy Group. **Usando dados para identificar como atuam as organizações de TI transformadas e como elas estão promovendo a vantagem digital.** 2018. Disponível em: <https://www.delltechnologies.com/pt-br/whitepaper/ esg-it-transformation-maturity-report-agility-innovation-business-value.htm>. Acesso em 10 mar. 2020.

FERREIRA, Filipa; SANTOS, Nélia; ANTUNES, Mário. **Clusters de alta disponibilidade– abordagem OpenSource**. Departamento de Engenharia Informática – ESTG, Leiria - Portugal, 2005.

FILIPPETTI, Marco Aurélio. **CCNA 5.0 – guia completo de estudo**. Florianópolis: Visual Books, 2014.

FOROUZAN, Behrouz A., MOSHARRAF, Firouz. **Redes de Computadores:** Uma Abordagem Top-Down. Porto Alegre: AMGH, 2013.

GUERRA, Paulo Asterio de Castro. **Uma abordagem arquitetural para tolerância a falhas em sistemas de software baseados em componentes**. Tese (Doutorado em Ciências da Computação) - Universidade Estadual de Campinas, Instituto de Matemática, Estatística e Computação Científica, Campinas, SP. 2004. 177p.

GUTIERREZ, M. A. **Sistemas de informação hospitalares: progressos e avanços.** J. Health Inform. 2011;3(2):I-II.

KOREN, Israel; KRISHNA, C. Mani. **Fault-tolerant systems.** Elsevier, 2010.

KOSCIANSKI, André; SOARES, Michel dos Santos. **Qualidade de Software:** Aprenda as metodologias e técnicas mais modernas para o desenvolvimento de software. 2.ed. São Paulo: Novatec Editora, 2007.

LAMBERSON, Jim. **Single and Multistage Watchdog Timers**. Sensoray. Retrieved, v. 10, 2013.. Disponível em: <https://sensoray.com/downloads/appnote_826_watchdog_1.0.0.pdf>. Acesso em: 07 mai. 2020.

LYU, Michael R. **Software reliability engineering: A roadmap**. In: Future of Software Engineering (FOSE'07). IEEE, 2007. p. 153-170. Disponível em <https://ieeexplore.ieee.org/abstract/document/4221618>. Acesso em: 07 mai. 2020.

MAGALHÃES, Ivan Luizio; PINHEIRO, Walfrido Brito. **Gerenciamento de Serviços de TI na Prática:** Uma abordagem com base na ITIL. São Paulo: Novatec Editora, 2008.

MARIN, H. F. **Sistemas de informação em saúde: considerações gerais.** J.Health Inform. 2010;2(1):24-8.

MORAIS, Alexandre Fernandes de. **Redes de computadores**. 1. Ed. São Paulo: Érica, 2014.

MORAIS, Alexandre Fernandes de. **Redes de computadores:** fundamentos. 7. Ed. São Paulo: Érica, 2010.

OLIVEIRA, Natália Q. de; SALLES, Ronaldo M. Uso de SDN no Balanceamento de Cargas em Redes com Suporte a Múltiplos Caminhos. In: SIMPÓSIO BRASILEIRO DE TELECOMUNICAÇÕES, SBrt2016, 2016, Santarém.

PEREIRA, Samáris Ramiro et al. **Sistemas de Informação para Gestão Hospitalar**. Journal of Health Informatics, v. 4, n. 4, 2012. Disponível em : <http://www.jhi-sbis.saude.ws/ojs-jhi/index.php/jhi-sbis/article/view/206>. Acesso em 16 Set. 2020.

SEBRAE, Serviço Brasileiro de Apoio às Micro e Pequenas Empresas Pesquisa Sebrae. **A Tecnologia da Informação e Comunicação (TIC) nas MPE Brasileiras**. 2015. Disponível em: < https://bibliotecas.sebrae.com.br/chronus/ARQUIVOS_CHRONUS/bds/bds.nsf/79461b2314b6d80a40a76844eea985bf/$File/5981.pdf > . Acesso em 09 mar 2020.

SHOOMAN, Martin L. **Reliability of computer systems and networks**: fault tolerance, analysis, and design. John Wiley & Sons, 2003.

SILVA, Jorge Cleber Pereira; AMARAL , Maria Fernanda Brito; NASCIMENTO, Anderson Lopes; FELIX, Iana Celia. **O impacto da Tecnologia da Informação na Gestão de Pequenas Empresas**. Revista Formadores - Vivências e Estudos, Cachoeira - Bahia, v. 12, n. 6, p. 47-60, out. 2019.

SOMASUNDARAM, G. et al. **Armazenamento e Gerenciamento de Informações:** Como armazenar, gerenciar e proteger informações digitais. Porto Alegre: Bookman, 2011.

WEBER, Taisy Silva. **Tolerância a falhas:** conceitos e exemplos. Apostila do Programa de Pós-Graduação–Instituto de Informática-UFRGS. Porto Alegre, p. 24, 2003. Disponível em: <http://www.inf.ufrgs.br/~taisy/disciplinas/textos/ConceitosDependabilidade.PDF>. Acesso em 10 mar. 2020.

WEBER, Taisy Silva. **Um roteiro para exploração dos conceitos básicos de tolerância a falhas**. Relatório técnico, Instituto de Informática UFRGS, 2002. Disponível em: <http://www.inf.ufrgs.br/~taisy/disciplinas/textos/Dependabilidade.pdf >. Acesso em 10 mar 2020.

XAVIER, L. H.; CARVALHO, T. C. M. B. **Gestão de resíduos eletroeletrônicos**: uma abordagem prática para a sustentabilidade. Rio de Janeiro: Campus, 2014.