

NÃO PODE FALTAR

ASPECTOS DE PROJETO DOS SISTEMAS DISTRIBUÍDOS

Caique Silva Pereira

QUAIS SÃO OS ASPECTOS MAIS RELEVANTES DE UM PROJETO DE IMPLANTAÇÃO DE UM SISTEMA DISTRIBUÍDO?

Em um projeto de implantação de um sistema distribuído é necessário definir alguns objetivos como: abertura, concorrência, escalabilidade, heterogeneidade, segurança, tolerância a falhas e transparência, alguns deles merecem uma atenção especial.



Fonte: Shutterstock.

Deseja ouvir este material?

Áudio disponível no material digital.

PRATICAR PARA APRENDER

Caro estudante, nesta seção, exploraremos com maior detalhamento alguns dos principais aspectos de projeto em sistemas distribuídos, nomeadamente: segurança, escalabilidade, resiliência e heterogeneidade, que são diferenciais em relação aos sistemas puramente de rede, mas que não sejam sistemas distribuídos.

Você já pensou em como os serviços de *streaming* de vídeo mais populares da atualidade conseguem suportar uma quantidade enorme de acessos simultâneos? Ainda, quando não há tanta demanda assim, será que empresas de grande porte mantêm ligados centenas de servidores mesmo que não estejam em uso? Já pensou no consumo de energia dessas máquinas ociosas? É exatamente nesta seção que você notará que entender sobre escalabilidade, neste caso, ajuda as empresas responsáveis por esses serviços a maximizarem seu faturamento, evitando uso desnecessário de recursos, sem comprometer a usabilidade e a consequente satisfação de seus clientes. Curioso para descobrir como? Então, daremos seguimento aos nossos estudos.

Você esteve no papel de um arquiteto de sistemas de uma empresa que estava desenvolvendo um sistema distribuído para controle de manutenção preventiva de veículos e identificou os principais objetivos que esse sistema deve atingir. Agora, é o momento de avançar no projeto, pesquisando e apresentando *frameworks*

atuais, utilizados por grandes empresas no segmento de serviços em TI, e garantir que o sistema esteja apto a ser comercializado. Prepare um relatório e uma apresentação com exemplos e descrições desses *frameworks*. É sempre bom unir a teoria estudada com as práticas do mercado de trabalho, para que sua formação seja o mais completa possível e, desta forma, você esteja preparado para pleitear as melhores oportunidades.

CONCEITO-CHAVE

Os sistemas distribuídos, normalmente, possuem alguns objetivos para atender determinados requisitos. Alguns são considerados desafios importantes que os profissionais da área devem considerar. Segundo Tanenbaum e Steen (2008), esses objetivos são:

- Abertura.
- Concorrência.
- Escalabilidade.
- Heterogeneidade.
- Segurança.
- Tolerância a falhas.
- Transparência.

Entretanto, eles devem ser analisados mais como desafios a serem atingidos, uma vez que nem sempre o sistema distribuído conseguirá atingir todos de maneira integral.

A seguir, descreveremos a que se refere cada uma dessas metas na visão de Tanenbaum e Steen (2008), embora os itens Segurança, Escalabilidade, Tolerância a Falhas e Heterogeneidade mereçam uma atenção especial.

ABERTURA

No contexto de sistemas distribuídos, refere-se a quanto o sistema é modularizado ou, em outras palavras, quanto é fácil integrar e alterar tecnologias e *frameworks* sem que o sistema seja comprometido. Você já deve ter ouvido falar em micro serviço, não é verdade? Esse nome é a maneira atual (*hype*) de dizer que o sistema possui uma grande abertura. Importante observar que, independentemente do que o termo "abertura" possa passar, é uma coisa positiva em sistemas distribuídos. É importante ter em mente esse conceito, pois, à primeira vista, um sistema mais "aberto" parece ser uma coisa negativa, no sentido de estar mais vulnerável a falhas e, na verdade, em sistemas distribuídos, isso não é verdade. Segurança não tem relação nenhuma com essa abertura, mas isso é um assunto a ser detalhado em uma próxima oportunidade. Por exemplo, se o sistema utiliza tecnologias não-proprietárias, dizemos que a abertura desse tal sistema é maior.

CONCORRÊNCIA

Refere-se à capacidade de o sistema poder ser acessado e utilizado de maneira simultânea, concorrente (ao mesmo tempo), por vários usuários. Aqui, cabe novamente a ressalva de que, no contexto de sistemas distribuídos, o termo concorrência não tem uma conotação negativa (como ocorre, por exemplo, no comércio), apenas refere-se a um sistema que dá suporte a acessos simultâneos. Por exemplo, várias pessoas podem acessar um website de comércio eletrônico, certo? Pois bem, isso é um acesso concorrente.

ESCALABILIDADE

Escalabilidade é um termo comum em redes de computadores e está relacionado à capacidade de o sistema poder ser escalável, ou seja, ampliado ou reduzido para suportar, por exemplo, uma maior quantidade de acessos simultâneos ou realizar uma tarefa mais rapidamente (ao ampliarmos a capacidade de processamento desse sistema). Perceba como algumas metas podem estar inter-relacionadas nos sistemas distribuídos, como é o caso da escalabilidade e da concorrência. Por exemplo, quando uma empresa de games necessita aumentar a quantidade de servidores de um determinado jogo on-line para possibilitar uma maior quantidade de jogadores em uma partida *multiplayer*, ela está escalando o sistema (nesse exemplo, aumentando a escala dele).

EXEMPLIFICANDO

Imagine um site de comércio eletrônico no período de promoções, por exemplo, no dia da *Black Friday*. O sistema deverá ser escalável para permitir picos de acesso simultâneo, dessa forma, uma das maneiras é aumentar a quantidade de nós/servidores, para que não inviabilize o negócio, deixando o sistema indisponível.

Segundo Tanenbaum e Steen (2008), um sistema cujo desempenho aumenta com o acréscimo de hardware e software, proporcionalmente à capacidade acrescida, é chamado escalável.

É importante notar, entretanto, que um sistema dito escalável permite que se aumente ou diminua a quantidade de recursos. Você deve estar se perguntando: por que eu diminuiria a capacidade do meu sistema?

EXEMPLIFICANDO

Imagine a seguinte situação: você criou uma aplicação web que distribui conteúdo em vídeo para preparar estudantes para fazerem a prova do ENEM. Você roda essa aplicação, de maneira replicada, em um conjunto de servidores em nuvem de algum provedor de *cloud computing* conhecido do mercado, digamos, com dez nós. Apesar de a ideia ser excelente, você nota que a quantidade de usuários que utiliza sua plataforma cai drasticamente entre os meses de novembro e junho, uma vez que os estudantes, normalmente, começam a se preparar para esse exame – que ocorre anualmente entre outubro e novembro – a partir de julho, quando estão de férias. Supondo que você paga para esse provedor de *cloud computing* R\$

150,00 mensais, para que este disponibilize os dez nós de maneira contínua. Não seria interessante que, nos meses de menor demanda, você diminuísse a quantidade de servidores pela metade, por exemplo, e pagasse a quantia de R\$ 75,00 mensais nesse período? Nesse cenário, sua economia seria de R\$ 600,00, que você poderia investir em outros projetos. Esse é um exemplo típico de escalabilidade “para baixo”.

Dentre os termos relacionados ao escalonamento, temos dois muito utilizados: escalonamento horizontal e escalonamento vertical.

ASSIMILE

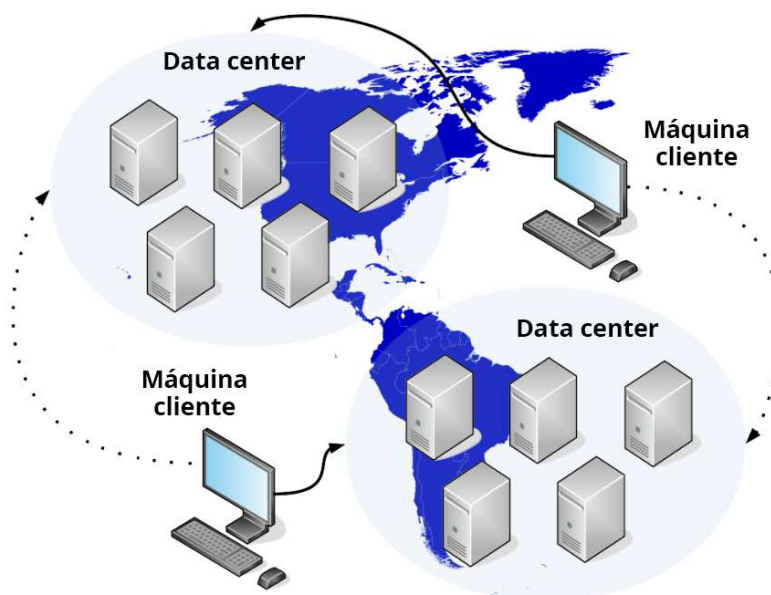
O **escalonamento horizontal** possibilita o aumento de máquinas, tais como servidores, assim há uma divisão da carga de trabalho entre eles, não sobrecarregando um servidor específico. No sistema *e-commerce*, em que temos os picos de acessos dos clientes simultaneamente, esse tipo de escalonamento é muito utilizado em sistemas distribuídos e em nuvem.

Já o **escalonamento vertical** permite aumentar recursos de processamento, como CPU e memória em máquinas existentes. Quando se deseja maior desempenho, ou na ocorrência de falhas de uma CPU, essa pode ser uma abordagem importante.

Dois aspectos importantes a serem levados em consideração em relação à escalabilidade são em termos geográficos e administrativos.

Escalabilidade em termos geográficos refere-se ao sistema, o qual, apesar de se apresentar como único para o usuário, está rodando em várias réplicas, em dois ou mais data centers geograficamente distintos. Podemos, por exemplo, utilizar um determinado provedor de *cloud computing* que possua data centers no estado de São Paulo, no Brasil, e no estado do Arizona, nos EUA. A Figura 3.28 ilustra tal cenário.

Figura 3.28 | Exemplo de escalabilidade geográfica



Fonte: elaborada pelo autor.

O benefício desse tipo de configuração é fornecer uma melhor experiência, em termos de conectividade e latência (atrasos na rede), para os usuários. Aqueles que estão mais próximos do Hemisfério Norte podem acessar a aplicação através dos data centers nos EUA, enquanto os usuários do Hemisfério Sul podem acessar a aplicação através dos data centers no Brasil. Outra vantagem é que, na ocorrência de um desastre, por exemplo, um furacão passar pelo Arizona, que comprometa o data center, todos os usuários poderão acessá-lo de outra localidade, incluindo os usuários mais próximos ao data center atingido (ainda que a usabilidade, do ponto de vista desses usuários, seja ligeiramente comprometida, devido a uma maior distância desse data center).

Escalabilidade em termos administrativos, refere-se ao escopo administrativo, que é afetado pela escalabilidade geográfica e é um conceito bastante simples de ser compreendido, embora muitas vezes ignorado. Imagine que, no cenário da Figura 3.28, os links de comunicação do lado dos EUA sejam fornecidos por provedores de internet daquela região, ao passo que os links de comunicação no lado do Brasil sejam fornecidos por provedores de internet daqui. Caso o link no lado dos EUA fique indisponível, não adiantará entrar em contato com o provedor de internet daqui do Brasil, pois é uma empresa diferente da que fornece o serviço nos EUA, administrativamente falando. Portanto, o escopo administrativo foi inerentemente ampliado, o que significa que o responsável pelo sistema distribuído terá mais trabalho para administrá-lo, incluindo, por exemplo, a necessidade de abrir um chamado de suporte técnico em outro idioma.

EXEMPLIFICANDO

A escalabilidade pode ser alterada, isto é, aumentada ou diminuída, de maneira automatizada, sem a intervenção do desenvolvedor, por meio de ferramentas, como Chef e Ansible. Podemos exemplificar a função de ambas as ferramentas em um cenário de um portal de notícias, que tem, durante a madrugada, baixa demanda de acessos. As ferramentas exercem o papel de diminuir automaticamente os recursos utilizados. Em outro exemplo, quando sai uma notícia muito esperada e os acessos ao portal sobem drasticamente, as ferramentas exercem o papel de aumentar os recursos disponibilizados para garantir o bom funcionamento.

HETEROGENEIDADE

O termo heterogeneidade vem de heterogêneo, ou seja, algo desigual, que apresenta estrutura, função ou distribuição diferente. Portanto, aplicando isso ao conceito de computação, quando temos heterogeneidade, estamos falando de um sistema que contenha em sua composição máquinas (nós) de sistemas operacionais, recursos (hardware) e até mesmo fabricantes diferentes. Esse é um dos aspectos mais frequentes de um sistema que utiliza arquitetura distribuída. Geralmente, o sistema é composto por máquinas de diversas características diferentes que se comunicam para manter o funcionamento.

Dessa forma, o sistema distribuído é capaz de funcionar em uma arquitetura heterogênea, tanto em termos de hardware quanto de software. Em termos de hardware, significa que o sistema distribuído consegue operar em nós com características de hardware diferentes, por exemplo, entre máquinas com diferentes valores de memória principal (RAM), memória de armazenamento (HD, SSD, etc.) e capacidade de processamento (*clock* dos processadores). Em termos de software, significa que o sistema distribuído suporta, por exemplo, diferentes sistemas operacionais, com alguns nós utilizando MS-Windows, enquanto outros utilizam alguma distribuição GNU/Linux. Isso pode parecer difícil de ser alcançado, mas existe um elemento que facilita nosso trabalho, o *middleware*, o qual, como o nome sugere, é uma camada de software que fica situada entre a sua aplicação e o sistema operacional. Não se preocupe se esse termo e seu papel não estiverem claros ainda, pois detalharemos esse assunto em uma aula posterior. Por exemplo, quando um website roda em máquinas com diferentes sistemas operacionais ou características de hardware (por exemplo, uma máquina do tipo servidor que possui mais memória RAM que outra máquina dentro do mesmo sistema distribuído), dizemos que se trata de um sistema distribuído heterogêneo.

Os protocolos de redes são fundamentais para que essa comunicação entre máquinas diferentes ocorra, porém, na maioria das aplicações distribuídas, precisamos também de um *middleware*.

O *middleware* pode ser considerado um conjunto de padrões e funcionalidades que atua como uma camada central entre a nossa plataforma, o sistema operacional e as nossas aplicações. Essa camada central permite que em um sistema distribuído rodem diferentes aplicações em diferentes plataformas e que todas elas consigam se comunicar adequadamente

Existem vários *frameworks* utilizados para implementação de plataformas de *middleware*, e os mais conhecidos são:

- **CORBA:** *framework* de implementação de *middleware* baseado na linguagem de programação C++.
- **.NET Remoting:** *framework* de implementação de *middleware*, baseado no ambiente de programação .NET, que pode utilizar várias linguagens diferentes, por exemplo, Visual Basic, C#, entre outras.
- **Akka.NET:** *framework* de implementação de *middleware*, baseado no ambiente de programação .NET, que pode utilizar várias linguagens diferentes, como Visual Basic, C#, entre outras.
- **Java RMI:** *framework* de implementação de *middleware*, baseado na linguagem de programação JAVA.
- **JAX-WS:** *framework* de implementação de *middleware*, baseado na linguagem de programação JAVA.

SEGURANÇA

Sem dúvida, um dos aspectos mais importantes no projeto de sistemas distribuídos é a segurança. Tipicamente, seja qual for a aplicação desenvolvida, sendo um sistema distribuído, funcionará em uma plataforma com várias máquinas, chamadas de nós, que replicam tal aplicação e, conforme já sabemos, a comunicação entre essas máquinas sempre ocorre por meio de redes de comunicação, tipicamente cabeadas.

É muito importante sempre considerar aspectos de segurança no projeto de sistemas distribuídos. Segundo Coulouris *et al.* (2013), em termos de sistemas distribuídos, podemos pensar em dois níveis: o da confidencialidade e o da integridade dos dados. A confidencialidade dos dados refere-se ao acesso ao dado por indivíduos ou sistemas não autorizados, e a integridade dos dados refere-se, além do acesso, à modificação do dado.

Conforme os autores, os pontos de atenção em relação à segurança no projeto de sistemas distribuídos são:

- **Portas são expostas:** sistemas distribuídos são construídos com base em um conjunto de processos que oferecem serviços e compartilham informação. As portas de comunicação pelas quais esses serviços se comunicam são, intrinsicamente, abertas (para que clientes possam acessar tais serviços) e, dessa forma, um hacker pode enviar mensagem a qualquer uma delas.
- **Redes de computadores não são seguras:** remetentes de mensagens podem ser falsificados e endereços IP podem estar duplicados, de forma que alguém malicioso possa receber as mesmas mensagens de um destinatário válido.
- **A validade das chaves criptográficas deve ser limitada:** quanto mais tempo uma mesma chave estiver válida e ativa, maiores são as chances de estar comprometida, por ter maiores chances de ser conhecida (e explorada) por uma quantidade maior de pessoas e sistemas.
- **Algoritmos de criptografia podem ter falhas:** na atualidade, a melhor prática é divulgar publicamente os algoritmos de criptografia para que a comunidade e as entidades especialistas possam validá-lo e sugerir melhorias, de forma que a privacidade esteja garantida pela chave criptográfica, e não pela inacessibilidade ao algoritmo utilizado.
- **Hackers podem ter acesso a recursos poderosos:** o custo dos recursos computacionais tem diminuído cada vez mais, de forma que máquinas poderosas estão acessíveis para a maioria da população. Assim, sempre considere que ataques podem ocorrer de inúmeras fontes e podem explorar vulnerabilidades, utilizando, inclusive, ataques do tipo força-bruta (que tentam descobrir senhas por tentativa e erro, com simples “chutes”).

TOLERÂNCIA A FALHAS

Tolerância a falhas refere-se à capacidade de o sistema distribuído se autorrecuperar na ocorrência de uma ou mais falhas. As falhas em um sistema computacional podem ocorrer em vários momentos ao longo do uso de um

sistema, por uma série de razões, as quais serão detalhadas mais adiante. Aproveitando, você sabia que, no nosso idioma, existe uma palavra para expressar essa ideia de “capacidade do sistema distribuído se autorrecuperar na ocorrência de uma ou mais falhas”? Isso mesmo, apenas uma palavra: resiliência. Então, da próxima vez que quiser passar essa ideia de maneira mais concisa, utilize-a.

Quando falamos de aspectos no projeto de sistemas distribuídos, temos que falar de resiliência de processos, pois é um dos principais objetivos a serem atingidos em uma aplicação distribuída. A resiliência de processos está relacionada com o sistema ter uma comunicação confiável entre as camadas de cliente e servidor. Sua ideia básica é de que os processos da nossa aplicação sejam replicados em grupos, isso faz com que o sistema tenha uma proteção contra falhas relacionadas a processos.

Para conseguir criar projetos tolerantes a falhas, temos que ter em nossos sistemas uma detecção de falhas, assim como conseguir mascarar todas as falhas apresentadas e a replicação de nosso sistema, para que ela seja imperceptível. Só podemos atingir estas características de acordo com questões de projetos, e a mais importante, neste caso, é verificar qual grupo de processos que a nossa aplicação deverá conter. Para isso, entenderemos o que são os grupos simples e os grupos hierárquicos e como esses processos são associados a eles.

Quando falamos de resiliência de processos, organizamos em grupos os processos que são considerados idênticos. Portanto, quando uma mensagem é enviada a um grupo de nosso sistema, a ideia é que todos os processos membros desse grupo recebam a mensagem. Se ocorrer uma falha em um dos processos no tratamento da mensagem, outro processo desse grupo deve tratar a mensagem no lugar do processo com falha.

Em um sistema distribuído, temos grupos dinâmicos, ou seja, os processos podem se mover entre os grupos que são criados. Um processo de um cadastro de usuário do site XYZ pode enviar uma mensagem para a realização do cadastro a um grupo de servidores sem precisar quantos existem e quem são eles.

■ GRUPO SIMPLES E GRUPO HIERÁRQUICO

Quando falamos de **grupos simples**, todos os processos são iguais e todas as decisões são tomadas entre todos os processos, ou seja, de forma coletiva. A grande vantagem deste tipo de grupo é que não há um ponto único para falha. Mesmo que ocorra falha ou caia algum processo, o grupo continua mantendo o serviço em funcionamento. Portanto, a execução do sistema não é centralizada em um só ponto.

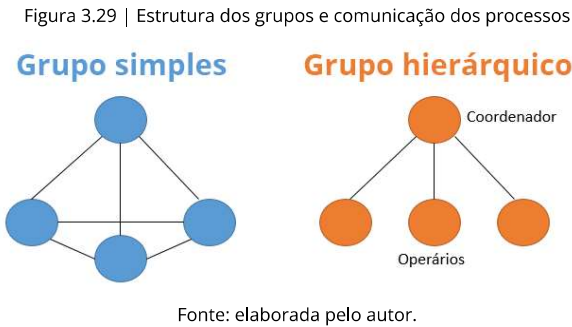
A desvantagem que encontramos nesse tipo de grupo é que a tomada de decisão tende a demorar mais, porque cada decisão deve ser priorizada pelos processos, tendo uma votação antes da tomada de decisão.

Já os **grupos hierárquicos**, como o nome mesmo já diz, são baseados em uma hierarquia, logo existem processos considerados mais importantes e que controlam toda execução. Nesses grupos, temos um processo chamado

“coordenador”, e os demais chamamos de “operários”. Sempre que chega uma nova requisição no sistema, ela é enviada ao processo coordenador, que decide o melhor operário para executá-la.

A grande vantagem desse tipo de grupo é que as decisões são centralizadas, portanto temos mais agilidade na tomada de decisão, o que gera um retorno mais rápido. Já a grande desvantagem apontada é que, caso ocorra uma falha no processo coordenador, o serviço todo para.

Podemos observar, na Figura 3.29, a estrutura dos grupos apresentados e como seus processos se comunicam.



Podemos observar que, na comunicação simples, os processos se comunicam entre si e decidem em conjunto qual é o processo mais adequado para executar determinada ação durante o funcionamento de um sistema. Já na comunicação hierárquica, os processos estão divididos entre “coordenador” e “operários”, em que o processo coordenador definirá o processo operário mais adequado para executar determinada ação durante o funcionamento de um sistema.

TRANSPARÊNCIA

Por fim, transparência refere-se, novamente, ao ponto de vista do usuário, o quão transparente o sistema é, ou seja, quanto o cliente “desconhece” do funcionamento interno do sistema, e isso é uma característica positiva: quanto menos o usuário necessitar saber da implementação e do funcionamento do sistema, ou seja, quanto mais transparente o sistema for, melhor para o usuário.

Na Tabela 3.1, podemos observar os tipos de transparência, segundo Tanenbaum e Steen (2008):

Tabela 3.1 | Tipos de transparência

Tipo	Descrição
Acesso	Ocultar diferenças na representação de dados e no modo de acesso.
Localização	O local do recurso é desconhecido.
Migração e relocação	O recurso pode ser movido, inclusive enquanto está em uso.

Tipo	Descrição
Replicação	Múltiplas instâncias de um recurso podem ser utilizadas para aumentar a confiabilidade e o desempenho.
Concorrência	Vários usuários podem compartilhar o mesmo recurso.
Falha	A falha e a recuperação de um recurso não afetam o sistema.

Fonte: elaborada pelo autor.

Com o conteúdo estudado nesta seção, você está apto a entender os aspectos mais relevantes de um projeto de implantação de um sistema distribuído. Percebeu como é importante definir esses objetivos em um projeto de sistemas distribuídos? Agora, continuaremos os estudos e seguiremos em frente com mais conceitos de sistemas distribuídos.

FAÇA A VALER A PENA

Questão 1

Quando planejamos implantar um sistema distribuído, há vários fatores importantes que devem ser levados em consideração. Podemos chamá-los de aspectos de projeto, e os principais são: segurança, escalabilidade, resiliência e heterogeneidade.

A capacidade de máquinas com diferentes sistemas operacionais se comunicarem na execução de um sistema se relaciona ao aspecto de:

a. Segurança.

b. Escalabilidade.

c. Resiliência.

d. Heterogeneidade.

e. Transparência.

Questão 2

Um dos aspectos mais importantes no projeto de sistemas distribuídos é a segurança. Tipicamente, seja qual for a aplicação desenvolvida, sendo um sistema distribuído, esta funcionará em uma plataforma com várias máquinas, chamadas de nós, que replicam tal aplicação. Essa plataforma tem a comunicação entre as máquinas feita através de redes de comunicação.

Sabendo que a segurança é um aspecto muito importante, analise as afirmativas a seguir com pontos de atenção em relação à segurança do sistema:

I. Máquinas de diferentes sistemas operacionais se comunicam.

II. Portas abertas.

III. Algoritmos de criptografia podem ter falhas.

IV. Camada de *middleware*.

V. A validade das chaves criptográficas.

Em relação aos pontos de atenção relacionados à segurança, são corretas as afirmativas:

a. I e IV, apenas.

b. I e II, apenas.

c. II, III e V, apenas.

d. II e V, apenas.

e. I e III, apenas.

Questão 3

O termo escalabilidade trata de outro aspecto do projeto de sistemas distribuídos. Um sistema, cujo desempenho aumenta com o acréscimo de hardware e software, proporcionalmente à capacidade acrescida, é chamado escalável. Esse termo está relacionado diretamente ao desempenho da aplicação e ao consumo de seus recursos.

Sobre o aspecto do projeto de escalabilidade em um sistema distribuído, analise as afirmativas a seguir e marque V para as verdadeiras ou F para as falsas:

- () Um sistema escalável é aquele que está disponível para acesso 100% do tempo.
- () Espera-se que um sistema distribuído escalável possa aumentar ou diminuir a quantidade de seus recursos.
- () Escalabilidade em termos geográficos refere-se ao sistema que está rodando em várias réplicas, em dois ou mais data centers geograficamente distintos.
- () O termo escalabilidade está relacionado à comunicação de máquinas e aos processos na execução do sistema.

Assinale a alternativa que representa a sequência correta:

a. V - V - V - F.

b. F - V - V - V.

c. V - F - V - F.

d. F - F - V - F.

e. F - V - V - F.

REFERÊNCIAS

AKKA.NET. Disponível em: <http://bit.ly/2NpB5Ja>. Acesso em: 18 fev. 2021.

COULOURIS, G.; DOLLIMORE, J.; KINDBERG, T.; BLAIR, G. **Sistemas Distribuídos:** conceitos e projeto. 5. ed. Porto Alegre: Bookman, 2013.

CORBA. Disponível em: <http://bit.ly/2ZHvZKD>. Acesso em: 18 fev. 2021.

JAVA RMI. Disponível em: <http://bit.ly/3qQtQZ1>. Acesso em: 18 fev. 2021.

JAX-WS. Disponível em: <http://bit.ly/2ZLjei0>. Acesso em: 18 fev. 2021.

TANENBAUM, A. S.; STEEN, M. V. **Sistemas Distribuídos**: princípios e paradigmas. 2 ed. São Paulo: Pearson, 2008.