

# Sistemas Distribuídos



Anhanguera

**AVALIE**  
SUA PROFISSÃO

QUANDO APARECER EM SEU  
PORTAL UMA AVALIAÇÃO SOBRE  
SEU CURSO, RESPONDA:



NOTAS

**9 ou 10**

SIGNIFICA QUE VOCÊ INDICA

NOTAS

**7 ou 8**

SIGNIFICA QUE VOCÊ NÃO INDICA



Anhanguera



Anhanguera



# Anhanguera

## **Definição e exemplos de sistemas distribuídos**

Mesmo que você não saiba, hoje mesmo você já deve ter acessado um sistema distribuído. Você deve estar pensando: como assim? Ao abrir o navegador de sua preferência e acessar uma página de internet, você está usando um sistema distribuído. Essa simples ação rotineira em nosso dia a dia, por meio de um smartphone ou computador, utiliza um sistema distribuído, mas, afinal, o que é um sistema distribuído?



Um sistema distribuído é um conjunto de computadores que são interligados via rede, mas, para o usuário final das aplicações que são executadas através deles, aparenta ser um sistema único, como uma única máquina ou um único software (TANENBAUM; STEEN, 2008). Um de seus principais aspectos é que os computadores que fazem parte de sistemas distribuídos têm o funcionamento independente, ou seja, cada um age por si próprio e, muitas vezes, os sistemas e hardwares dessas máquinas são totalmente diferentes, porém aparentam ser uma coisa só para o usuário. Esses computadores estão ligados por rede e só assim é possível seu funcionamento de forma distribuída.

A principal motivação para construir e utilizar sistemas distribuídos é proveniente do desejo de compartilhar recursos. O termo “recurso” é bastante abstrato, mas caracteriza bem o conjunto de elementos que podem ser compartilhados de maneira útil em um sistema de computadores interligados em rede. Ele abrange desde componentes de hardware, como discos e impressoras, até entidades definidas pelo software, como arquivos, bancos de dados e objetos de dados de todos os tipos, conforme indica Coulouris et al. (2013).



Os sistemas distribuídos são, sem sombra de dúvida, mais utilizados em arquiteturas do tipo cliente-servidor e, como vimos anteriormente, esse tipo de arquitetura apresenta recursos compartilhados (tanto a nível de hardware quanto a nível de software) para permitir que milhares de clientes tenham acesso a esses recursos e possam utilizá-los como se houvesse uma comunicação direta entre as máquinas e o cliente.





# Anhanguera

## **Computação em cluster e computação em grid**

Os sistemas distribuídos podem ser classificados em diferentes categorias, de acordo com sua arquitetura e finalidade, sendo os mais comuns computação em cluster e computação em grid. Agora vamos falar um pouco sobre algumas características da computação em cluster. Esse tipo de computação é formado por um conjunto de máquinas com hardware semelhantes, ou seja, as máquinas que compõem o cluster possuem características homogêneas, de acordo com Tanenbaum e Steen (2008). O conjunto de máquinas que compõem o cluster são ligadas por rede local (LAN).



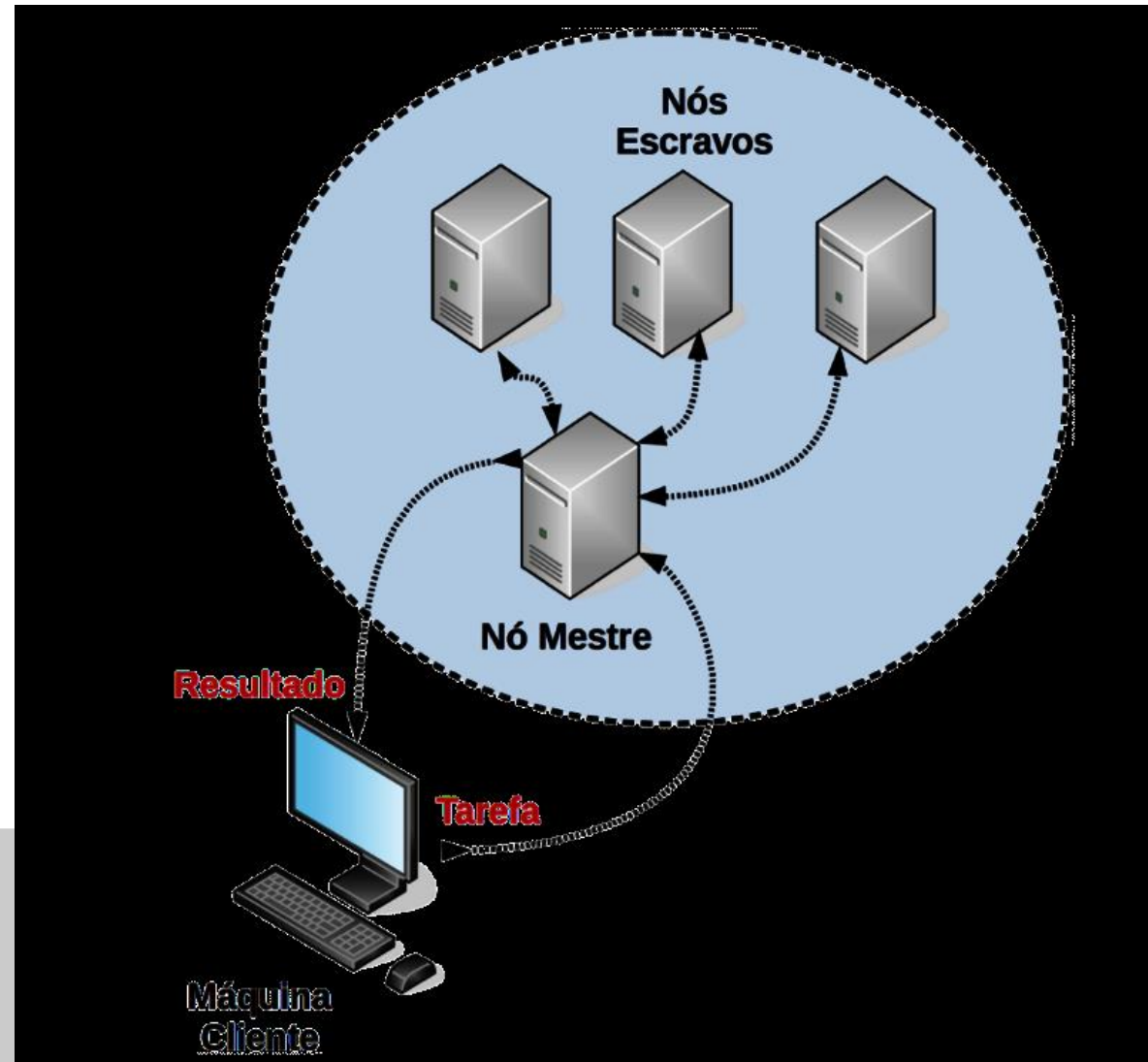
# Anhanguera

Quando falamos da parte de software da computação em cluster, temos algumas características importantes. Na maioria das vezes, o sistema operacional entre as máquinas que formam o cluster é equivalente. Além disso, é frequente que um único programa funcione de forma paralela, ou seja, um programa é subdividido em partes menores e cada parte é executada em uma máquina (ou nó) desse cluster, de forma distribuída, a fim de obter um aumento significativo de desempenho e, conseqüentemente, executar determinada tarefa em menos tempo. Geralmente as máquinas desse tipo de sistema são fortemente acopladas (conforme definição apresentada na Seção 1.2) em suas ligações, muitas vezes podem até compartilhar a mesma memória RAM entre várias máquinas. Há sempre uma das máquinas que chamamos de nó mestre, ou seja, a máquina principal que gerencia o funcionamento da aplicação entre todos os nós. O nó mestre faz a interface com o usuário, aloca tarefas e administra a fila de tarefas. A Figura 1.10 ilustra um cluster, que se encontra dentro do círculo pontilhado.





# Anhanguera







# Anhanguera

## **Computação em grid**

Agora que já conhecemos a computação em cluster, vamos falar da computação em grid, ou grades, e apontar algumas características. Conforme Tanenbaum e Steen (2008), esse tipo de computação é formado por um conjunto de máquinas com características diferentes, podendo o hardware e os sistemas operacionais ser de fabricantes diferentes. Com isso, temos uma característica heterogênea na computação em grid. Essencialmente um sistema de computação em grid interliga vários clusters. Um exemplo de grid é o CineGrid, que trabalha no desenvolvimento de ferramentas colaborativas multimídia (CINEGRID, 2018). Na Figura 1.11, vemos uma parte desse grid, com as ligações entre clusters no Brasil; mas saiba que o CineGrid interliga outros centros de pesquisa, em várias partes do mundo.



# Anhanguera





# Anhanguera

## **Diferenças entre clusters e grids**

Muitas vezes, pode parecer que clusters e grids são a mesma coisa, mas existe uma característica fundamental que difere esses dois tipos de sistemas distribuídos. Para facilitar o entendimento das diferenças entre eles, podemos pensar que clusters são sistemas homogêneos, ou seja, são criados para executar alguma tarefa específica que, em geral, necessita de um alto poder de processamento e, portanto, levaria muito tempo para ser executado em um computador convencional.



# Anhanguera

## **Exemplificando**

Um exemplo de tarefa que leva muito tempo para ser executada é o treinamento de redes neurais artificiais de aprendizagem profunda (deep learning) para uso em ferramentas de chat on-line. Nesse tipo de aplicação, os chamados bots – que, nesse caso, são o resultado da rede neural treinada – conversam com e respondem questionamentos dos clientes (usuários) do serviço de chat online de uma determinada empresa. Recentemente, muitos sistemas bancários têm utilizado esse tipo de recurso. Supondo que para executar o treinamento desse tipo de rede em computador convencional seriam necessários dois dias, ao executarmos essa tarefa em um cluster, esse treinamento poderia ser realizado em questão de minutos.



# Anhanguera

Para diminuirmos o tempo de processamento, poderíamos executar esse tipo de tarefa (treinamento de redes neurais artificiais de aprendizagem profunda) em um computador com alto poder de processamento, porém o custo desse tipo de computador pode tornar essa opção inviável. Segundo o Lawrence Livermore National Laboratory (2018), o Sequoia é utilizado para realizar simulações numéricas referentes à física de armas nucleares. Assim sendo, este computador é um cluster homogêneo, visto que executa uma pesquisa de finalidade específica.



# Anhanguera

Por sua vez, podemos pensar que os grids têm uma abordagem heterogênea, ou seja, são criados para executarem diferentes tarefas, de certa maneira relacionadas entre si, formando um centro de pesquisas de caráter multidisciplinar. Uma maneira ainda mais simples de entender essa característica é enxergar o grid como um conjunto de dois ou mais clusters, cada um deles responsável por um certo tipo de pesquisa.



# Anhanguera

## **Sincronização de Relógios**

Sistemas formados por múltiplos computadores necessitam sincronizar suas ações entre si, e uma das maneiras mais utilizadas, dada sua simplicidade e popularidade, é sincronização horária, por meio do protocolo conhecido como Network Time Protocol (NTP) (NTP, 2018). Esse protocolo, por sua vez, utiliza o protocolo de transporte de dados User Datagram Protocol (UDP), operando na porta 123. Essencialmente esse protocolo é utilizado para sincronização do relógio das máquinas locais (desktops, notebooks, servidores) e demais dispositivos de rede.

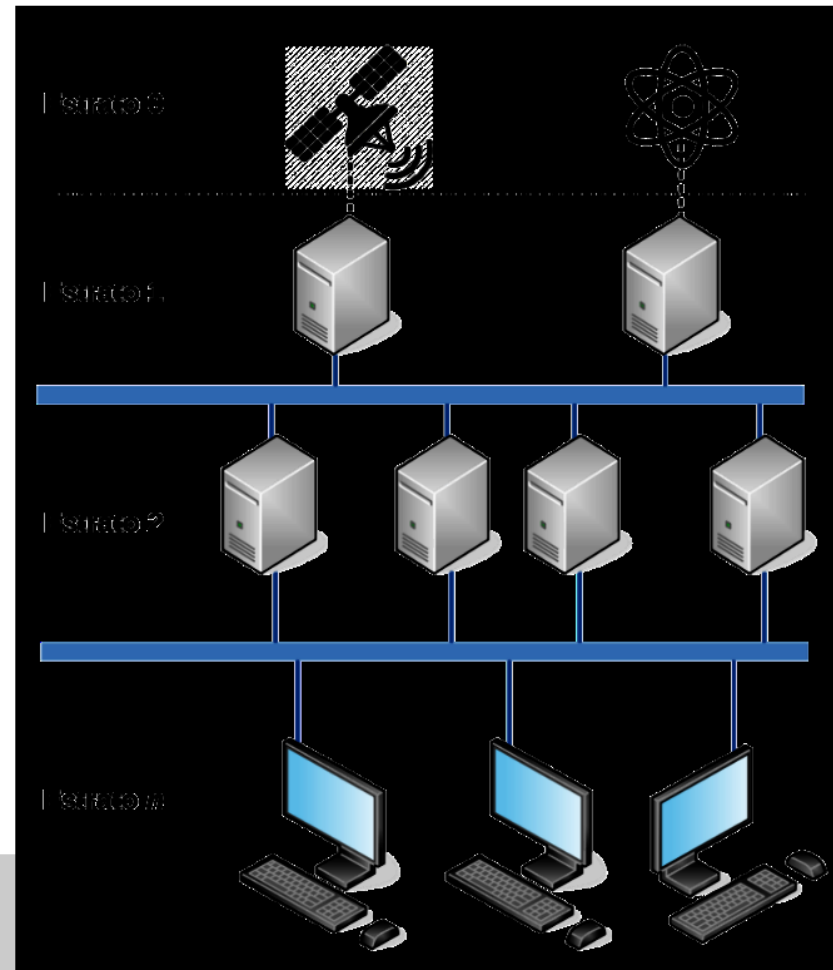




A referência horária é dada por sistemas de altíssima precisão, como os relógios atômicos (NTP, 2018). Dada a precisão desses sistemas, computadores conectados a eles pertencem a uma camada de servidores chamada de estrato 1 (os sistemas de alta precisão em si pertencem a uma camada topológica chamada de estrato 0). Segundo NTP (2018), como não existem muitos servidores no mundo conectados diretamente a relógios atômicos, outros servidores são conectados aos de estrato 1, que por sua vez formam uma segunda camada de servidores de horário, chamada de estrato 2, e essa hierarquia se estende até os servidores de estrato 15, conforme podemos ver na Figura 1.12. Os computadores dos usuários são configurados para atualizar a informação horária por meio da rede, consultando servidores de estratos com valores mais altos.



# Anhanguera





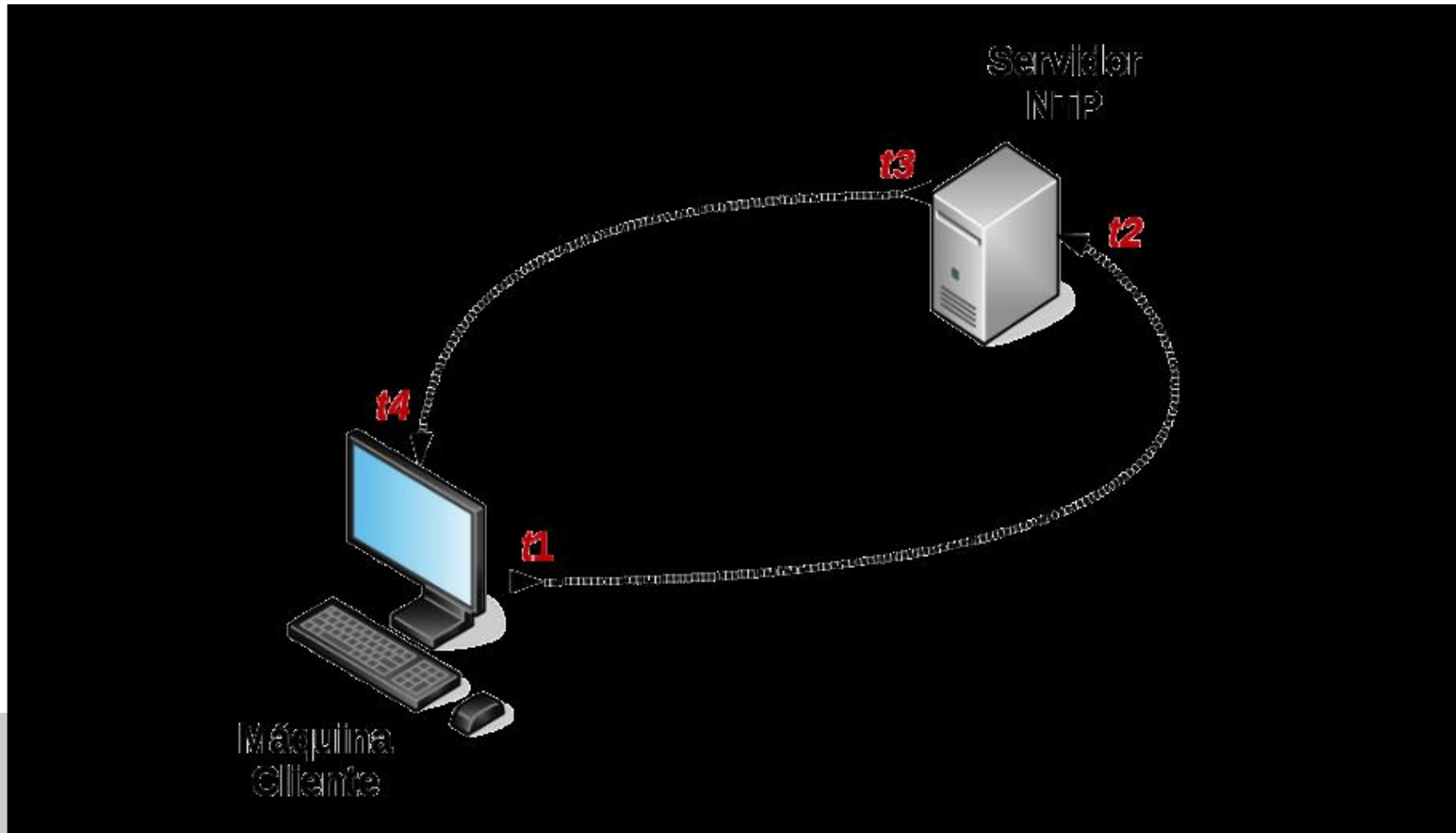
# Anhanguera

Aqui no Brasil, por exemplo, temos o Observatório Nacional (ON) que é o Órgão oficial responsável pela geração, conservação e disseminação da Hora Legal Brasileira.

Um aspecto muito interessante do protocolo NTP é que ele é projetado para verificar a latência (atraso, delay) entre a máquina cliente e a máquina servidora, e a implementação disso é na verdade muito simples: essencialmente, de tempos em tempos, a máquina cliente faz uma consulta a um servidor NTP para verificar em quanto seu relógio está atrasado (ou adiantado) em relação ao servidor horário de referência, processando a seguinte operação:  $(t_4 - t_1) - (t_3 - t_2)$ , em que  $t_1$  é a hora, minuto, segundo e milésimos de segundo (também chamada de timestamp) da máquina cliente ao enviar uma requisição (através de um pacote, no contexto de redes de computadores) para o servidor NTP;  $t_2$  é o timestamp do servidor ao receber essa requisição;  $t_3$  refere-se ao timestamp em que um pacote de resposta a essa requisição é enviada ao cliente; e  $t_4$  é o timestamp em que o cliente recebe a resposta do servidor NTP. Esse cálculo, portanto, resulta em quanto o sistema operacional deverá atrasar (ou adiantar) o relógio da máquina local para que ela esteja sincronizada com a referência horária em questão, por exemplo, a Hora Legal Brasileira, em nosso caso. Um exemplo desse processo é ilustrado na Figura 1.13.



# Anhanguera





Alguns serviços, como os de acesso remoto e de autenticação de usuários, podem não funcionar adequadamente caso haja uma diferença muito grande no horário da máquina cliente (solicitante) em relação à máquina servidora (que roda e disponibiliza o serviço). Por esse motivo, é muito importantesaber como habilitar a sincronização horária das máquinas utilizando o NTP.

A configuração, bem como suas etapas, varia entre os sistemas operacionais e entre suas versões. Para realizar essa configuração em um sistema operacional Windows 10, siga o procedimento abaixo:

1. Abra o Prompt de Comando (CMD), que pode ser acessado pelo menu iniciar, ou pelo campo de busca, digitando prompt ou cmd.
2. Na janela do CMD, vamos inserir o código a seguir e pressionar a tecla “Enter” (Figura 1.14):

```
w32tm /config /syncfromflags:manual /manualpeerlist:0.pool.ntp.org
```



```
CA: Administrador: Prompt de Comando  
Microsoft Windows [versão 10.0.17134.285]  
(c) 2018 Microsoft Corporation. Todos os direitos reservados.  
  
C:\WINDOWS\system32>w32tm /config /syncfromflags:manual /manualpeerlist:0.pool.ntp.org  
O comando foi concluído com êxito.  
  
C:\WINDOWS\system32>
```

Na Figura 1.14, podemos observar que ao final do comando aparece a mensagem “O comando foi concluído com êxito”. Lembre-se: para executar os comandos, você deve estar como um usuário Administrador ou executar o CMD como Administrador. Agora que fizemos o apontamento para o servidor NTP que vamos utilizar, devemos reiniciar o serviço de data e hora para aplicar as alterações:

3. Vamos utilizar o comando `net stop w32time` e `net start w32time` para parar o serviço e iniciar, reiniciando-o, conforme podemos observar na Figura 1.15:



C:\> Administrador: Prompt de Comando

```
C:\WINDOWS\system32>net stop w32time  
O serviço de Horário do Windows está sendo finalizado .  
O serviço de Horário do Windows foi finalizado com êxito.
```

```
C:\WINDOWS\system32>net start w32time  
O serviço de Horário do Windows está sendo iniciado.  
O serviço de Horário do Windows foi iniciado com êxito.
```





Após isso, vamos forçar uma sincronização de data e hora para verificar se está tudo funcionando corretamente.

4. Podemos forçar uma sincronização por meio do comando `w32tm /resync /rediscover`, conforme observamos na Figura 1.16.

```
Administrator: Prompt de Comando

C:\WINDOWS\system32>w32tm /resync /rediscover
Enviando comando resync para o computador local
O comando foi concluído com êxito.

C:\WINDOWS\system32>
```



Sistemas formados por múltiplos computadores, como os sistemas distribuídos, necessitam sincronizar suas ações entre si, e uma das maneiras mais utilizadas, dada sua simplicidade e popularidade, é a sincronização horária, a qual é necessária para o funcionamento da maioria das aplicações.

Para fazer a sincronização horária entre os computadores que foram os nossos sistemas distribuídos, utilizamos um protocolo muito popular. Identifique a alternativa que corresponde ao protocolo utilizado para sincronização horária.

- a) Transmission Control Protocol – TCP.
- b) Network Time Protocol – NTP.
- c) User Datagram Protocol – UDP.
- d) HyperText Transfer Protocol – HTTP.
- e) SSH Remote Protocol – SSH.



Quando trabalhamos com sistemas distribuídos, há um nó que chamamos de nó mestre. Esse nó tem uma grande importância no funcionamento de nosso sistema e pode ser considerado o nó principal de um sistema que funciona com a arquitetura distribuída.

Considerando as características do nó mestre da computação em cluster, analise as opções abaixo e escolha a que não corresponde a uma tarefa no nó mestre.

I – Interface entre servidores e usuários.

II – Controlar a fila de tarefas entre os servidores.

III – Fazer o balanceamento das aplicações entre os outros nós.

IV – Interface entre os usuários.

V – O nó mestre recebe tarefas de outros nós para execução do sistema.

a) A opção I não é uma função do nó mestre.

b) A opção II não é uma função do nó mestre.

c) A opção III não é uma função do nó mestre.

d) A opção IV não é uma função do nó mestre.

e) A opção V não é uma função do nó mestre.