

**Aula 00 - Somente em  
PDF**

*SEFAZ-AM (Analista de Tecnologia da  
Informação da Fazenda) Sistemas  
Distribuídos*

Autor:  
**Evandro Dalla Vecchia Pereira**

05 de Janeiro de 2023

# Índice

1) Sistemas Distribuídos .....	3
2) Questões Comentadas - Sistemas Distribuídos - Multibancas .....	10
3) Servidores de Aplicação JEE (Java Enterprise Edition) .....	17
4) Questões Comentadas - Servidores de Aplicação JEE - Multibancas .....	24
5) Servidor de Aplicação JBoss-Wildfly .....	30
6) Questões Comentadas - Servidor de Aplicação JBoss-Wildfly - Multibancas .....	36
7) Lista de Questões - Sistemas Distribuídos - Servidores de Aplicação JEE - Multibancas .....	41



# SISTEMAS DISTRIBUÍDOS

Vamos começar o assunto com a definição de sistemas distribuídos, segundo dois autores conhecidos. Silberschatz define um sistema distribuído como:

“Uma coleção de processadores que não compartilham a memória ou um clock. Em vez disso, cada processador possui sua própria memória local. Os processadores se comunicam entre si através de diversas redes de comunicações, como barramentos de alta velocidade”.

Tanenbaum define que: “Um sistema distribuído é um conjunto de computadores independentes entre si que se apresenta a seus usuários como um sistema único e coerente.”

Resumindo, são sistemas que **o usuário “enxerga” como um único computador, mas na verdade existem alguns (ou muitos), geograficamente distribuídos**. Como as mensagens são trocadas e como é realizado o processamento, o usuário nem fica sabendo (é transparente), ele só recebe o resultado final.

Imagine uma situação em que são necessários diversos cálculos para resolver um problema, como por exemplo, verificar se há sinais enviados através de estrelas ou planetas (vida extraterrestre). Então, quem quiser participar desse projeto, pode ceder processamento de seu computador quando estiver ocioso (através de um programa previamente instalado). Os computadores voluntários realizam os cálculos e enviam resultados a outros computadores. Parece loucura, não? Mas existe esse projeto, é só verificar em <<https://www.seti.org/>>.

Note que o exemplo acima, que é real, mostra que para um usuário se trata de um único sistema, mas “por baixo dos panos” diversos computadores interagem, realizam cálculos etc. Pode ser que um computador possua processador RISC, outro CISC, um com sistema operacional Unix, Linux, Windows, Mac OS, e por aí vai. No fim, todos se entendem (de forma **transparente**) e o usuário vê apenas uma interface.

Quando falamos em transparência, não é apenas em relação à localização. Vamos ver na tabela a seguir que um sistema distribuído pode requerer alguns tipos de transparência.

Transparência	Descrição
Acesso	Oculto diferenças na representação de dados e no modo de acesso a um recurso.
Localização	Oculto o lugar em que um recurso está localizado.
Migração	Oculto que um recurso pode ser movido para outra localização.
Relocação	Oculto que um recurso pode ser movido para outra localização enquanto em uso.
Replicação	Oculto que um recurso é replicado.
Concorrência	Oculto que um recurso pode ser compartilhado por diversos usuários concorrentes.
Falha	Oculto a falha e a recuperação de um recurso.



Os sistemas distribuídos são muito comuns em ambientes que exigem alta disponibilidade, pois mesmo, que parte dele “caia”, o sistema ainda continua operando. Em comparação a um sistema centralizado, as **principais vantagens** são a **possibilidade de crescimento incremental** e a **possibilidade de implementação de tolerância a falhas** através da replicação de processos em unidades de computação distintas.

Vejamos alguns exemplos de sistemas distribuídos bastante conhecidos:

Buscadores:



Internet:



Computação em Nuvem:



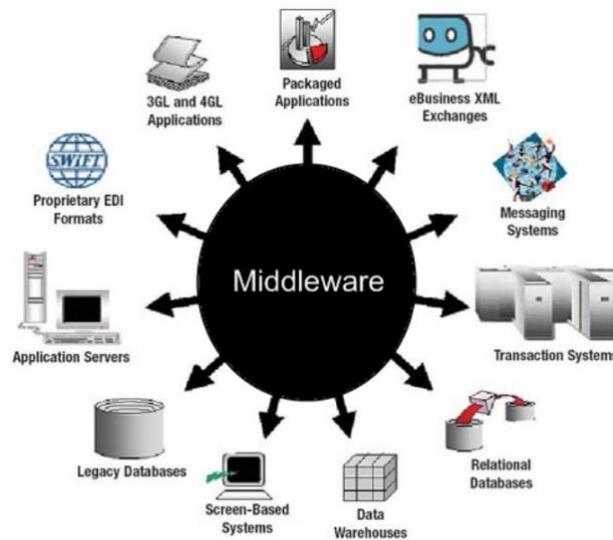
Alguns conceitos recorrentes em provas de concurso são:

**Middleware:** camada de software localizada logicamente entre uma camada de nível mais alto (usuários e aplicações) e uma camada subjacente (frameworks e facilidades de comunicação). As principais funções da camada de middleware são:

- Ocultar do usuário que a aplicação é executada em diferentes máquinas distribuídas geograficamente;



- Ocultar a heterogeneidade dos sistemas operacionais (ex.: Windows, Linux etc.) e protocolos de comunicação subjacentes.



**Processo:** conjunto dos recursos alocados a uma tarefa para sua execução. Um processo é um programa em execução ou uma forma de gerenciar recursos. Cada tarefa precisa de um conjunto de recursos para executar e atingir o seu objetivo: processador, memória, dados, arquivos, conexões de rede etc.

**Escalonamento:** em muitas situações pode haver mais de um processo competindo pelo uso de um mesmo recurso. O escalonador do sistema distribuído é o responsável em decidir o nó de processamento responsável pelo processamento da tarefa. Portanto, trata-se de um dos componentes mais importantes, o qual utiliza um algoritmo de escalonamento.

As **principais características** de um sistema distribuído são:

- Concorrência: diferentes nós do sistema distribuído executam-se em concorrência;
- Não há relógio global: os relógios locais não estão necessariamente iguais, assim a única forma de coordenação é por troca de mensagens;
- Falhas independentes: qualquer componente pode falhar de forma independente de outros

As **vantagens** de sistemas distribuídos incluem a possibilidade de seu crescimento incremental (computadores e linhas de comunicação podem ser acrescidos ao sistema), a possibilidade de implementação de aplicações inerentemente distribuídas e a possibilidade de implementação de tolerância a falhas e a replicação de processos em unidades de computação distintas.

As principais **motivações** para a utilização de um sistema distribuído podem ser:

- Distribuição geográfica (organização com instalações em várias localidades diferentes);
- Necessidade de ligação entre organizações independentes;
- Necessidade de extensibilidade, modularidade ou crescimento gradual;
- Partilha de recursos como troca de informação entre departamentos e/ou empresas;
- Necessidade de maior disponibilidade ou replicação de dados;
- Necessidade de maior desempenho ou de distribuição de carga.



Parece que tem só coisa boa, heim? Não...vamos a alguns **desafios**:

- **Segurança:** intrusões podem acarretar na leitura de mensagens em trânsito e/ou a injeção de novas mensagens;
- **Escalabilidade:** o sistema deve continuar a funcionar de forma eficaz mesmo que haja um crescimento significativo no número de recursos e no número de clientes;
- **Heterogeneidade:** Equipamentos com representações de dados diferentes (arquiteturas RISC, CISC), sistemas operacionais diferentes (Windows, Linux etc.) devem ser integrados no mesmo sistema distribuído.

## Funções, Modelos/Arquiteturas e Plataformas

**Gerenciamento de recursos:** a principal tarefa do sistema é garantir o controle sobre quem usa o quê. Esse controle também permite o compartilhamento dos recursos.

**Gerenciamento de processos:** essa funcionalidade distribui o processamento de forma justa entre as aplicações, evitando que uma aplicação monopolize o recurso. A gerência do processador está diretamente relacionada ao escalonamento. A sensação de que um processador está sendo compartilhado entre os vários processos é dita pseudoparalelismo.

Nos sistemas distribuídos podemos observar o paralelismo real, típico dos sistemas multiprocessados. O ciclo de processamento típico pode ser resumido da seguinte maneira:

- **Requisitar:** se a requisição não pode ser atendida imediatamente, então o processo requisitante deve esperar até obter o recurso;
- **Usar:** O processo pode operar sobre o recurso;
- **Liberar:** O processo libera o recurso.

**Modelo cliente-servidor:** os servidores mantêm recursos e servem pedidos de operações sobre esses recursos. Servidores podem ser clientes de outros servidores. Os próprios nomes ajudam a entender (servidor serve recursos e o cliente é quem solicita). Trata-se de uma arquitetura simples e permite distribuir sistemas centralizados muito diretamente, porém é pouco escalável, limitado pela capacidade do servidor e pela rede que o liga aos clientes. Ou seja, um único servidor não consegue atender milhares de clientes de forma satisfatória, por exemplo.

**Modelo peer-to-peer:** todos os processos possuem papéis semelhantes, sem distinção entre clientes e servidores. As principais características são mais ampla distribuição de carga (computação e rede), maior escalabilidade, o sistema expande-se acrescentando mais pares, e a coordenação mais complicada que cliente-servidor.

Componentes interagem entre si como “pares” por meio de chamadas/retorno numa rede. Não há hierarquia, como ocorre em um modelo cliente-servidor, e cada par pode interagir com qualquer outro. Exemplos de sistemas distribuídos em Peer-to-Peer (P2P) são abundantes, mas os atualmente mais conhecidos podem ser citados: BitTorrent, uTorrent, eMule, entre outros.



Vamos ver algumas **plataformas** que dão suporte aos sistemas e aplicações distribuídas:

- Java Messages: gestão de filas de mensagens da plataforma J2EE;
- MSMQ: sistemas de filas de mensagens da Microsoft;
- MQseries: sistemas de filas de mensagens da IBM;
- Active Enterprise: plataforma de integração da fabricante Tibco;
- Biztalk: Enterprise integration broker da Microsoft;

São inúmeras as soluções tecnológicas para implementar a distribuição:

- Utilização das interfaces de comunicação distribuída com *sockets*;
- Plataformas Cliente-Servidor;
- *Brokers* de Mensagens ou *Message Oriented Middleware*;
- Sistemas de Invocação Remota de Objetos (ex.: RMI ou CORBA);
- Sistemas Operacionais Distribuídos que permitam distribuição de todos os serviços do sistema;
- Web Services.

## Cluster x Grid

Agora vamos ver um tópico BASTANTE ABORDADO em provas de concurso, os **tipos de sistemas distribuídos**: computação em cluster, computação em grade (*grid*) e computação em nuvem. Vamos focar nos dois primeiros, a seguir.

**Clusters** são sistemas de processamento compostos por uma **coleção de computadores autônomos**, interconectados e que trabalham em conjunto para processar uma tarefa. Tipicamente na computação de cluster é utilizada programação paralela na qual um único programa é executado em paralelo. Dependendo do objetivo que se pretenda, podemos ter os seguintes tipos de cluster:

- Cluster de **alto desempenho**: direcionado a aplicações bastante exigentes no que diz respeito ao processamento;
- Cluster de **alta disponibilidade**: tem como objetivo permanecer ativo por um longo período e em plena condição de uso. Consegue detectar erros e se proteger de possíveis falhas;
- Cluster de **balanceamento de carga**: controla a distribuição equilibrada do processamento. As tarefas de processamento são distribuídas o mais uniformemente possível entre os nós.

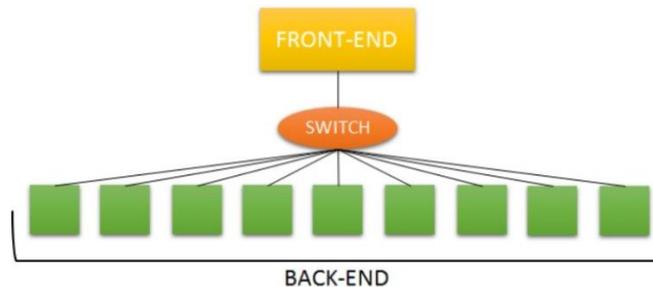
As **principais vantagens da utilização de clusters** são:

- custo-benefício: pode-se obter resultados tão bons quanto ou até superiores que um servidor sofisticado a partir de um conjunto de máquinas simples;
- escalabilidade: é possível aumentar a capacidade de um cluster com a adição de nós ou remover máquinas para reparos sem interromper a aplicação;
- facilidade de personalização para o atendimento da aplicação;



- existência de opções de softwares sem custos de licenciamento para cluster.

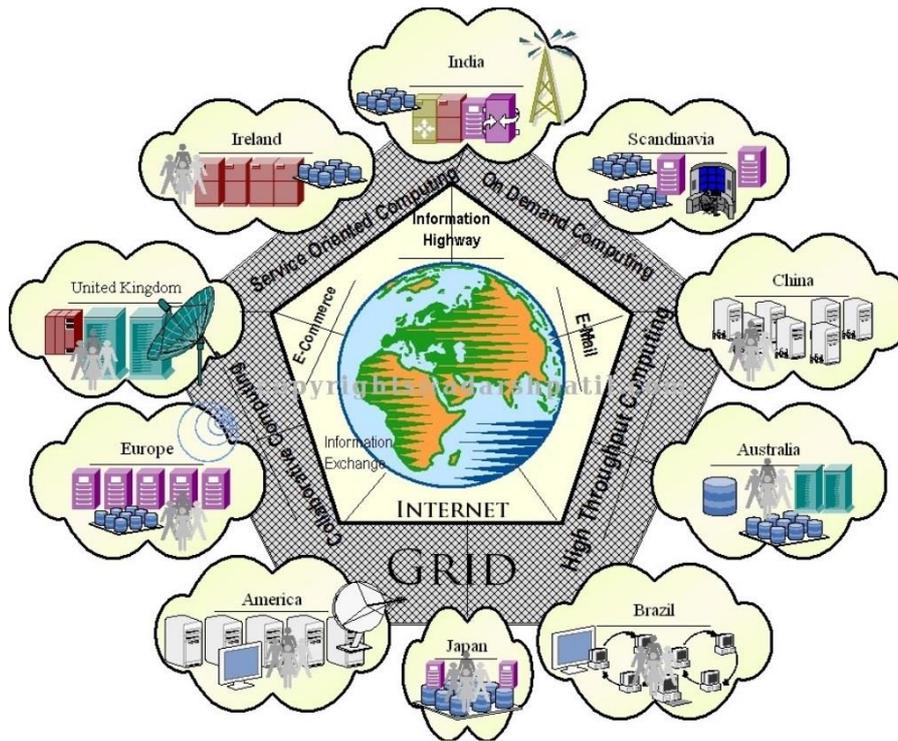
Resumindo, cluster é um agregado de computadores, um recurso de processamento utilizado em situações em que se exige a alocação exclusiva de um conjunto de recursos por longo período de tempo. Geralmente, existe um nó mestre que gerencia e divide as tarefas entre os demais nós (escravos).



Por exemplo, um software para tentar quebrar uma senha por força bruta pode ser paralelizado em um cluster, sendo que o mestre envia para cada escravo uma fatia das possibilidades de senha (um recebe a fatia de AAAA... até AZZZ..., outro recebe de BAAA... até BZZZ..., e assim por diante, se algum escravo descobrir a senha, avisa ao mestre para que tome as providências de cancelar as tarefas dos demais). Para finalizar o exemplo, esse cluster possui 8 computadores do mesmo modelo, sistema operacional Linux e o programa foi feito em C, com uma biblioteca MPI (para programação paralela).

Importante levar para o prova que um **cluster é um sistema fortemente acoplado**, ou seja, os computadores compartilham uma rede local, o mesmo sistema operacional em cada computador e há troca de mensagens entre eles. Note que foge um pouco do conceito geral de sistemas distribuídos, ficando mais voltado para um sistema paralelo. Mas para efeito de concurso, os clusters são cobrados junto com os grids (que veremos na sequência).

A **computação em grid** (grade – em português) possui uma estrutura de processamento descentralizada que não exige um tipo de controle central (não há um nó mestre), trabalhando em um ambiente cooperativo. Possui uma **estrutura fracamente acoplada**, com múltiplas máquinas interligadas através de uma rede de computadores. Por exemplo, milhares de máquinas no mundo, com sistemas operacionais diferentes, utilizadas para resolver cálculos para encontrar vida extraterrestre (já vimos nesta aula). Não há um requisito de alta disponibilidade dos equipamentos e a localização dos equipamentos é transparente para os usuários.



## QUESTÕES COMENTADAS

1. (CESPE/MBASA - 2010) Clusters ou combinações de clusters são usados quando os conteúdos são críticos, apesar de não haver necessidade de estarem disponíveis e (ou) processados rapidamente.

### Comentários:

Os clusters são estruturas cujo propósito é proporcionar alta disponibilidade, em situações nas quais há requisitos de criticidade de fornecimento do conteúdo, portanto há a necessidade que estejam disponíveis e geralmente se deseja um processamento rápido. Portanto, a questão está **errada**.

2. (IADES/PG-DF - 2011) Segundo Andrew Tanenbaum (2007) “Sistema Distribuído é uma coleção de computadores independentes que se apresenta ao usuário como um sistema único e consistente”. Assinale a alternativa correta a respeito de um sistema de informação distribuído.

A) A distribuição de tarefas se dá a partir de requisições do usuário, que indica o endereço do servidor onde deseja executar tal tarefa.

B) Em uma rede de computadores há servidores dedicados a atender pedidos dos clientes e estes, por sua vez, têm função exclusiva de requisitantes.

C) Todos os computadores de uma rede executam tarefas de cliente e servidor, quando se deseja integrá-los em uma arquitetura de sistemas distribuídos.

D) A transparência de acesso é uma característica dos sistemas distribuídos que permite que recursos sejam acessados sem que sua localização seja determinada.

E) Em um sistema de objetos distribuídos é possível invocar métodos de um objeto, ainda que este não esteja presente no computador do usuário.

### Comentários:

(A) A distribuição de tarefas indica a tarefa a ser desempenhada, sem indicar o endereço do servidor onde será executada tal tarefa. Lembre que um dos princípios de sistemas distribuídos é a transparência, portanto o usuário não tem ideia qual servidor será utilizado.

(B) Em um sistema distribuído não há a figura fixa dos servidores dedicados a atender requisições.

(C) Todos? Se for utilizada a arquitetura peer-to-peer até pode ser, mas quando é utilizada a arquitetura cliente-servidor?

(D) O conceito mostrado é o da transparência de localização, não de acesso! Vejamos:

- Transparência de acesso: oculta diferenças na representação de dados e no modo de acesso a um recurso.
- Transparência de localização: oculta o lugar em que um recurso está localizado.



(E) É isso aí! Em um sistema de objetos distribuídos é possível invocar métodos de um objeto remoto. Para tanto, podemos fazer uso de tecnologias como RMI (Remote Method Invocation) ou Corba.

Portanto, a **alternativa E** está correta e é o gabarito da questão.

**3. (FCC/TRE-PE - 2011) Arquitetura padrão proposta pelo Object Management Group (OMG) para estabelecer e simplificar a troca de dados entre sistemas distribuídos heterogêneos por meio de uma estrutura comum para o gerenciamento de objetos distribuídos que é conhecida como Object Manager Architecture (OMA). Trata-se de**

- A) IDL.
- B) RPC.
- C) DCON.
- D) CORBA.
- E) COM.

**Comentários:**

Uma das possibilidades de invocação de métodos remotos em sistemas distribuídos é através do uso de Corba ou RMI. RMI é mais utilizado, por ser mais simples que Corba. Corba é a arquitetura padrão proposta pelo Object Management Group (OMG) para estabelecer e simplificar a troca de dados entre sistemas distribuídos heterogêneos por meio de uma estrutura comum para o gerenciamento de objetos distribuídos. Portanto, a **alternativa D** está correta e é o gabarito da questão.

**4. (CESPE/SERPRO - 2013) Com relação à arquitetura de sistemas distribuídos, julgue os próximos itens. Na arquitetura distribuída, os sistemas orientados a eventos possuem processos fortemente acoplados.**

**Comentários:**

Na arquitetura distribuída, os sistemas orientados a eventos possuem processos fracamente acoplados. Portanto, a questão está **errada**.

**5. (FUNCAB/MDA - 2014) São desvantagens dos sistemas distribuídos:**

- A) dependência de hardware e custo maior de desenvolvimento.
- B) custo maior de desenvolvimento e maior probabilidade de ocorrência de falhas (bugs) no programa.
- C) dependência de sistemas operacionais e necessidade de maior processamento.
- D) necessidade de maior processamento e pouca disponibilidade.



E) ausência de controle das suas operações e dependência do tipo de rede utilizada para ligar as localidades.

#### Comentários:

(A) Sistemas distribuídos são heterogêneos (plataformas de hardware e sistemas operacionais distintos). (B) Sistemas distribuídos possuem maior complexidade de desenvolvimento (mais chance de bugs) e maior custo. (C) Sistemas distribuídos não dependem de sistemas operacionais específicos. (D) Sistemas distribuídos estão mais disponíveis. (E) Sistemas distribuídos permitem controle das operações e não dependem do tipo de rede utilizada para ligar as localidades. Portanto, a **alternativa B** está correta e é o gabarito da questão.

**6. (CESPE/TRE-PE - 2015) Os sistemas operacionais têm rotinas que não são executadas de forma linear, mas, sim, concorrentemente, em função de eventos assíncronos.**

#### Comentários:

As rotinas do sistema operacional são executadas conforme as características das tarefas (jobs). Existem os jobs do tipo CPU bound e os I/O bound. Os eventos variam conforme a necessidade dos usuários (mais uso do processador ou de dispositivos de entrada/saída). Por isso, a assertiva afirmou que as rotinas são executadas em função de eventos assíncronos! Portanto, a questão está **correta**.

**7. (CESPE/TJ-DFT - 2015) Nos sistemas implementados a partir do uso de uma arquitetura de componentes distribuídos, o middleware tem a responsabilidade de gerenciar a interação entre esses componentes.**

#### Comentários:

Middleware: camada de software localizada logicamente entre uma camada de nível mais alto (usuários e aplicações) e uma camada subjacente (frameworks e facilidades de comunicação). As principais funções da camada de middleware são:

- Ocultar do usuário que a aplicação é executada em diferentes máquinas distribuídas geograficamente;
- Ocultar a heterogeneidade dos sistemas operacionais (ex.: Windows, Linux etc.) e protocolos de comunicação subjacentes.

Portanto, a questão está **correta**.

**8. (CESPE/STJ - 2015) A arquitetura orientada a serviços é forma de desenvolvimento de sistemas distribuídos em que os componentes de sistemas são serviços autônomos, razão por que, devido à interoperabilidade, as ligações entre os serviços devem ser rígidas para não provocar mudanças durante sua execução.**

#### Comentários:



Não é o foco de nossa aula a arquitetura orientada a serviços. Mas como é um assunto pertinente, vamos ver... A arquitetura orientada a serviços é forma de desenvolvimento de sistemas distribuídos em que os componentes são autônomos e as ligações entre os serviços devem ser flexíveis durante sua execução. Portanto, a questão está **errada**.

**9. (IBFC/EBSERH - 2017) Assinale a alternativa correta. Cluster é um conceito que está diretamente relacionado aos sistemas de alta disponibilidade. Existem vários tipos de cluster, no entanto há alguns que são mais conhecidos, como:**

- A) cluster paralelo - cluster de alta disponibilidade - cluster de três camadas
- B) cluster de alto desempenho - cluster virtual - cluster matricial
- C) cluster paralelo - cluster virtual - cluster para balanceamento de carga
- D) cluster de alto desempenho - cluster matricial - cluster de três camadas
- E) cluster de alto desempenho - cluster de alta disponibilidade - cluster para balanceamento de carga

#### **Comentários:**

Clusters são sistemas de processamento compostos por uma coleção de computadores autônomos, interconectados e que trabalham em conjunto para processar uma tarefa. Tipicamente na computação de cluster é utilizada programação paralela na qual um único programa é executado em paralelo. Dependendo do objetivo que se pretenda, podemos ter os seguintes tipos de cluster:

- Cluster de alto desempenho: direcionado a aplicações bastante exigentes no que diz respeito ao processamento;
- Cluster de alta disponibilidade: tem como objetivo permanecer ativo por um longo período de tempo e em plena condição de uso. Consegue detectar erros e se proteger de possíveis falhas;
- Cluster de balanceamento de carga: controla a distribuição equilibrada do processamento. As tarefas de processamento são distribuídas o mais uniformemente possível entre os nós.

Portanto, a **alternativa E** está correta e é o gabarito da questão.

**10. (FGV/IBGE - 2017) Keyse é gerente de um centro de dados que hospeda distintos tipos de aplicação. Recentemente Keyse recebeu uma solicitação de hospedagem de uma aplicação de missão crítica, a qual requer uma disponibilidade de 99,8% ao ano.**

**Considerando V para as afirmativas verdadeiras e F para as falsas, analise as características do cluster para atender o requisito de alta disponibilidade do sistema.**

**( ) capacidade de distribuir igualmente todo o tráfego de entrada entre todos os nós do cluster, para evitar a sobrecarga de requisições a qualquer um deles e o consequente travamento do cluster;**

**( ) ter nós em espera, em quantidade suficiente, para assumir automaticamente a função de outro nó defeituoso;**



- ( ) realizar processamento da aplicação de forma paralela entre os vários nós do cluster;
- ( ) monitoramento dos nós feito por eles mesmos, por uma rede diferente da rede de dados.

A sequência correta é:

- A) V – F – V – F;
- B) F – V – F – V;
- C) V – V – V – F;
- D) V – V – F – F;
- E) F – F – V – V.

#### Comentários:

O foco do cluster é a alta disponibilidade, então vejamos:

(F) Seria verdadeira se o foco fosse o balanceamento de carga!

(V) Ter nós sobrando é importante para garantir a alta disponibilidade, no caso de falhas de outros

(F) O foco é manter o funcionamento (alta disponibilidade) e não paralelizar a execução dos programas (o que seria o foco do alto desempenho).

(V) O monitoramento dos nós feito por eles mesmos, sem depender de “alguém de fora” e a rede utilizada é separada da rede de dados.

Portanto, a **alternativa B** está correta e é o gabarito da questão.

**11.(CESPE/IPHAN - 2018) Cluster e GRID possuem características semelhantes, como o multiprocessamento distribuído, mas se distinguem pelas seguintes razões: cluster é uma coleção de máquinas paralelas conectadas entre si que, por trabalharem juntas, fornecem alta disponibilidade e(ou) balanceamento de carga; o GRID, por sua vez, envolve a integração, gestão de recursos computacionais fracamente acoplados e geograficamente distribuídos.**

#### Comentários:

Um belo resumo! Para o cluster pense em máquinas em uma mesma rede, numa mesma “sala”, são fortemente acoplados (mesmo sistema operacional, mesmo tipo de hardware). Os grids são espalhados pelo mundo, fracamente acoplados (sistemas operacionais e máquinas diferentes). Portanto, a questão está **correta**.

**12.(UFLA/UFLA - 2018) O aumento da facilidade de acesso à Internet tem permitido uma grande disponibilização da informação. Para dar suporte a essa facilidade de acesso é necessária uma enorme**



infraestrutura de hardware e software. Considerando as características de computadores paralelos, analise as proposições a seguir:

I. Um sistema de multiprocessamento simétrico pode ser composto por milhares de computadores com processadores e sistemas operacionais heterogêneos.

II. Um sistema de processamento paralelo em massa visa resolver problemas que exigem capacidade de utilização de memória compartilhada usando um único conjunto de núcleos de processamento localizados em um mesmo computador.

III. Um cluster de computadores é uma coleção de dois ou mais computadores usados para executar um dado problema podendo conter processadores multicore.

IV. Um grid computing provê uma plataforma na qual recursos computacionais são organizados dentro de um ou mais conjuntos lógicos, as tarefas são divididas entre diversos computadores locais ou remotos formando um “super computador virtual”.

Assinale a alternativa CORRETA:

- A) Somente as proposições I e II estão corretas.
- B) Somente as proposições II e III estão corretas.
- C) Somente as proposições III e IV estão corretas.
- D) Somente as proposições I e IV estão corretas.

#### Comentários:

(I) Não vimos nesta aula, mas não tem como garantir a simetria com milhares de computadores com processadores e sistemas operacionais distintos!

(II) Um sistema de processamento paralelo (cluster) utiliza troca de mensagens, pois cada computador tem sua memória. Ex.: o mestre envia tarefas aos escravos (via troca de mensagens), os escravos realizam as tarefas e retornam com os resultados. E por aí vai...

(III) Exato, um agregado de computadores (multicore ou não), utilizados para executar um dado problema (ex.: quebrar uma senha por força bruta).

(IV) Grid é fracamente acoplado, geralmente “espalhado pelo mundo”, tendo os recursos computacionais organizados dentro de um ou mais conjuntos lógicos. A ideia é possuir um “super computador virtual”.

Portanto, a **alternativa C** está correta e é o gabarito da questão.

**13.(INSTITUTO AOCP/PC-ES - 2019) Se um servidor Web estiver sobrecarregado, basta agregar novos servidores e dividir a carga de processamento. Considerando o funcionamento dos principais serviços de rede, como é conhecido o esquema de agregação de servidores?**



- A) SIP Apache.
- B) Divserver.
- C) Cluster.
- D) Upload.
- E) ARPANET.

**Comentários:** Clusters são sistemas de processamento compostos por uma coleção de computadores autônomos, interconectados e que trabalham em conjunto para processar uma tarefa. Tipicamente na computação de cluster é utilizada programação paralela na qual um único programa é executado em paralelo. Dependendo do objetivo que se pretenda, podemos ter os seguintes tipos de cluster:

- Cluster de alto desempenho: direcionado a aplicações bastante exigentes no que diz respeito ao processamento;
- Cluster de alta disponibilidade: tem como objetivo permanecer ativo por um longo período e em plena condição de uso. Consegue detectar erros e se proteger de possíveis falhas;
- Cluster de balanceamento de carga: controla a distribuição equilibrada do processamento. As tarefas de processamento são distribuídas o mais uniformemente possível entre os nós.

Portanto, a **alternativa C** está correta e é o gabarito da questão.

#### 14.(UFMG/UFMG - 2019) Um Cluster de Alta Disponibilidade é caracterizado por

- A) no máximo 2 servidores para redundância ou tolerância a falhas e fontes de energia redundantes, dentre outros dispositivos.
- B) diversos servidores para redundância ou tolerância a falhas e fontes de energia redundantes, dentre outros dispositivos.
- C) diversos servidores conectados por uma rede wifi e fontes de energia redundantes, dentre outros dispositivos.
- D) um único servidor com diversas fontes de energia redundantes ligados a um gerador para que esse servidor quase nunca desligue por falta de energia.

#### Comentários:

Cluster de alta disponibilidade: tem como objetivo permanecer ativo por um longo período e em plena condição de uso. Consegue detectar erros e se proteger de possíveis falhas.

Ou seja, deve haver diversos servidores e fonte de energia redundante (geradores, fornecimento de energia por mais de uma empresa etc.). Portanto, a **alternativa B** está correta e é o gabarito da questão.



# SERVIDORES DE APLICAÇÃO JEE (JAVA ENTERPRISE EDITION)

Java é uma plataforma completa de desenvolvimento e execução, sendo composta de três elementos: a máquina virtual Java (JVM), um conjunto de APIs (*Application Programming Interfaces*) e a linguagem de programação Java (bastante utilizada na atualidade). Também é importante deixarmos claro que o “universo” Java é composto basicamente de três famílias:

- Java SE (*Standard Edition*): é a base da plataforma, incluindo o ambiente de execução e as bibliotecas Java;
- Java ME (*Micro Edition*): voltada para o desenvolvimento de aplicações para dispositivos móveis e embarcados;
- Java EE (*Enterprise Edition*): voltada para o desenvolvimento de **aplicações corporativas e para internet**.

O nosso foco será a especificação JEE (*Java Enterprise Edition*), além de APIs importantes no contexto dos servidores de aplicação, entre outros componentes. Vamos lá...

## Especificação JEE

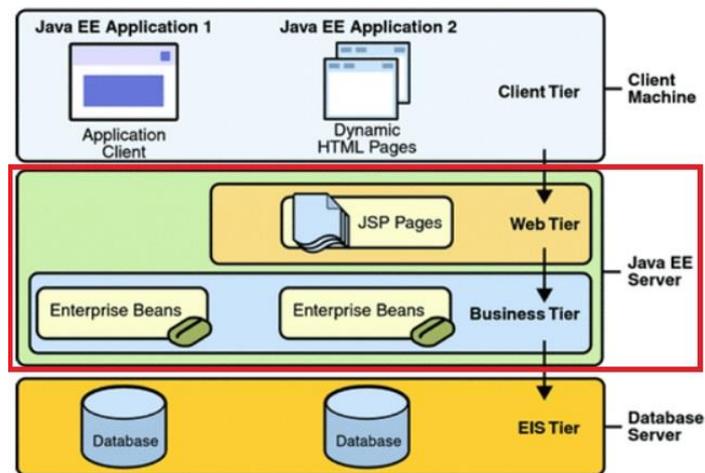
Ao estudarmos a especificação JEE (chamado antigamente de J2EE), veremos alguns conceitos, tais como servlets, Beans, JSP, containers etc., os quais são essenciais para um melhor entendimento do assunto. Mas o que é a JEE? Trata-se de um **conjunto de especificações destinadas a facilitar a criação de aplicações empresariais em Java**. Ou seja, é uma “super especificação” que abrange outras especificações, tais como servlets, JSP, EJB, JDBC, JPA, JTA, JNDI, JMS, entre outras.

JEE define uma especificação modelo para criar aplicações, no qual atividades comuns (persistência de dados, validações, tratamento de requisições HTTP etc.) são especificadas e padronizadas, para os usuários lerem, implementarem e usarem. Também são definidos aspectos como alta disponibilidade, *clustering*, mensageria, entre outros. Além disso, ele também dispõe de APIs e *frameworks* (plataformas de desenvolvimento) para aplicações JEE escaláveis.

Se cada aplicação a ser criada houvesse a necessidade de implementar novamente aspectos como acesso a dados, segurança, entre outros, a vida dos administradores e dos programadores seria um caos! Certo? Por isso, essa é a proposta da especificação JEE, definir uma estrutura padronizada, necessárias às aplicações corporativas, de forma **reutilizável para reduzir o retrabalho**.

JEE disponibiliza bibliotecas que fornecem funcionalidades para implementar software Java distribuído, tolerante a falhas e multicamada, baseada amplamente em componentes modulares, executados em um servidor de aplicações. A figura abaixo ilustra a arquitetura de um servidor de aplicações JEE (destaque em vermelho). Ainda nesta aula veremos as outras especificações (JSP, Beans etc.).





## APIs JEE

A especificação JEE define uma série de APIs, para facilitar a vida do programador. Vamos focar em algumas que possuem relação com o conteúdo desta aula:

- **JDBC (Java Data Base Connectivity):** conjunto de interfaces escritas em Java que faz o **envio de instruções SQL para banco de dados**. Possibilita o uso de bancos de dados instalados, sendo que para cada banco de dados há um driver JDBC;
- **JPA (Java Persistence API):** *framework* para persistir objetos Java simples (POJOS - *Plain Old Java Objects*). Entre as principais características da JPA temos as consultas em objetos da *Java Persistence Query Language (JPQL)*;
- **JTA (Java Transaction API):** especifica interfaces locais que são utilizadas entre um gerenciador de transação<sup>1</sup> e as partes que estão envolvidas em uma transação, como a aplicação, o gerenciador de recursos e o servidor de aplicação;
- **JNDI (Java Naming and Directory Interface):** API para acesso a serviços de nomes e diretórios, permitindo que aplicações cliente descubram e obtenham dados ou objetos através de um nome (semelhante ao serviço DNS, na Internet). A JNDI é utilizada em aplicações Java que acessam recursos externos, como base de dados, filas, entre outros;
- **JMS (Java Message Service):** uma especificação e um sistema de mensageria para servidores de aplicação JEE. Seus principais componentes são *JMS Provider*, *JMS Clients* e *Administered Objects*.

Há situações em que as aplicações precisam se comunicar de forma assíncrona com outras aplicações ou com um componente dentro do mesmo aplicativo. Por exemplo, algumas operações podem ser registradas

---

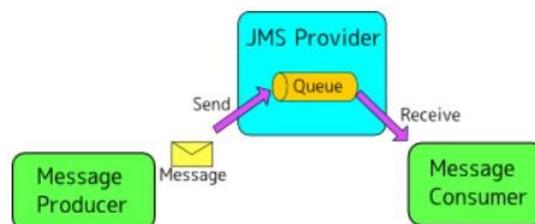
<sup>1</sup> Transação: sequência de operações que são tratadas como um bloco único e indivisível e segue basicamente quatro princípios: atomicidade, consistência, isolamento e durabilidade.

e pode ser preferível que o log seja escrito de forma assíncrona, em vez de aguardar que a transação seja concluída. Os sistemas de mensageria surgiram para resolver essa questão.

Mensagens são uma forma de comunicação entre softwares distribuídos de forma fracamente acoplado. Em comunicação por mensagens, o remetente envia uma mensagem ao destinatário, porém ambos não precisam estar disponíveis ao mesmo tempo (assim como funciona um e-mail). O remetente não precisa saber nada sobre o destinatário, precisa saber apenas qual o formato da mensagem e qual o destino.

A lógica da aplicação permite que seja enviada uma mensagem e ela continue operando sem receber uma resposta imediatamente. Alguns conceitos recorrentes quando tratamos de serviços de mensageria são:

- *Producer/Publisher JMS*: cria e envia mensagens;
- *Consumer/Subscriber JMS*: recebe mensagens;
- *Destination*: objeto que o cliente usa para especificar o alvo das mensagens que produz e a origem de mensagens que consome;
- *JMS Queue*: fila ou área contendo mensagens que aguardam para serem lidas. As mensagens não são necessariamente lidas na ordem que a fila propõe;
- *JMS Topic*: mecanismo de publicação de mensagens para múltiplos leitores;
- *Publish/Subscribe*: uma mensagem que é enviada a um conjunto de destinatários;
- *Point to point (Queue)*: uma mensagem que é enviada a um único destinatário.



## Servlets

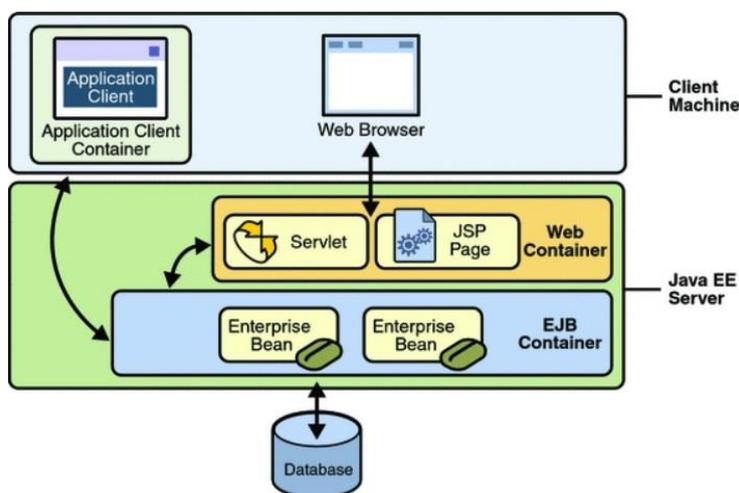
Servlets são utilizados para o desenvolvimento de aplicações Web com conteúdo dinâmico. Contém uma API que abstrai e disponibiliza os recursos do servidor Web (ex.: Apache, IIS etc.) de maneira simplificada para o programador. Resumindo, servlet é basicamente uma classe na linguagem de programação Java que dinamicamente processa requisições e respostas, proporcionando novos recursos aos servidores.

As aplicações baseadas em servlet geram conteúdo dinâmico (geralmente HTML) e interagem com os clientes, utilizando o modelo request/response, normalmente através do protocolo HTTP. Um Servlet necessita de um container Web para ser executado.



## EJB (Enterprise Java Beans)

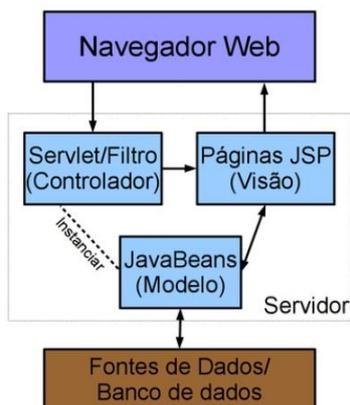
O EJB (*Enterprise JavaBeans*) é um dos principais componentes da plataforma JEE, pois nele é implementada a **lógica de negócio**. É um componente do tipo servidor que executa no container do servidor de aplicação. EJBs permitem que o desenvolvedor se concentre nas necessidades do negócio do cliente, enquanto questões de infraestrutura, segurança, disponibilidade e escalabilidade ficam sob responsabilidade do servidor de aplicações. EJBs são utilizados no desenvolvimento de componentes de software, e normalmente são encapsulados no container EJB:



Os principais objetivos da tecnologia EJB são fornecer um rápido e simplificado desenvolvimento de aplicações Java baseado em componentes distribuídos, transacionais, seguros e portáteis.

## JSP (Java Server Page)

JSP (*Java Server Page*) é uma extensão da tecnologia Servlet. Trata-se de um documento convertido pelo container JSP em um servlet. JSP permite que os programadores Web possam desenvolver conteúdos dinâmicos com a reutilização de componentes predefinidos com interação de componentes que utilizam script do lado do servidor. JSP é uma especialização do servlet que permite que conteúdo dinâmico seja facilmente desenvolvido. Ou seja, enquanto o foco do EJB é em regras de negócio, o foco da JSP é na apresentação:



Fazendo um comparativo, JSP é similar às tecnologias Active Server Pages (ASP) da Microsoft e ao PHP, ou seja, é utilizada no desenvolvimento de aplicações dinâmicas para Web. Uma página criada com a tecnologia JSP, após instalada em um servidor de aplicação compatível com a tecnologia JEE, é transformada em um servlet.

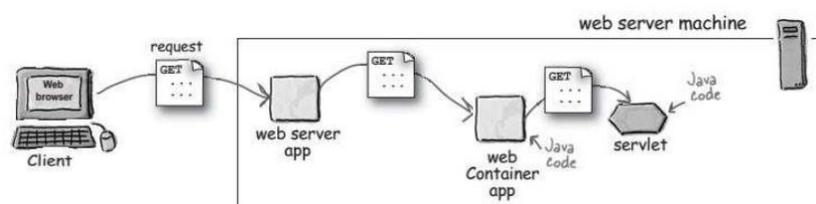
## Container JEE

Um container é um objeto que contém outros objetos, no contexto da especificação JEE. Podemos pensar em um “big objeto” que possui outros objetos, associando ao nome “container”, fica mais fácil para lembrar. Esses objetos podem ser incluídos ou removidos dinamicamente, em tempo de execução, diferentemente do que ocorre com uma composição onde esse relacionamento é fixado em tempo de compilação. Um container é uma interface entre um componente e uma funcionalidade de baixo nível de uma plataforma.

Os cinco containers JEE são:

- *JEE Server*: é o container mais completo que implementa a especificação *JEE server*, provê containers Servlet, EJB e EJB;
- Container EJB: gerencia a execução de Enterprise Beans (os modelos de negócio);
- Container Web: gerencia a execução de páginas JSP (aquelas dinâmicas) e componentes servlet para aplicações Java EE;
- *Application client* container: gerencia a execução dos componentes da aplicação cliente. Aplicações clientes e seus containers rodam no cliente;
- Servlet container: gerencia a execução de servlets. O container gerencia o ciclo de vida, dá suporte ao *multithread*, segurança, e suporte para páginas JSP, no caso dos containers web.

Quando uma aplicação Web recebe uma requisição para um servlet, o servidor Web (o Apache, por exemplo) a encaminha não diretamente ao servlet, mas para o container do servlet:

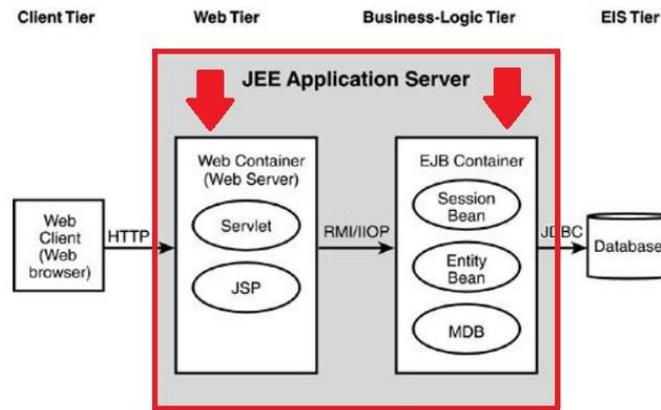


Em JEE, um container pode conter e gerenciar componentes como Servlets (servlet container - para aplicações Web) ou EJBs (EJB container - para componentes de negócio).

## Servidores de Aplicação JEE

Servidores de aplicação JEE completam a especificação JEE, e disponibilizam uma plataforma de middleware baseada em padrões abertos, em conformidade com a especificação JEE. Um servidor de aplicação JEE é também chamado servidor *full compliance*. Eles implementam o web container e o EJB container:





Assim, um servidor de aplicações totalmente compatível com a especificação JEE (*full compliance*) engloba tanto um container web, como um container EJB. Por exemplo, o Tomcat não é um servidor de aplicações JEE full, pois implementa apenas Servlets e JSP.

E qual é o conceito de servidor de aplicações? Servidores de aplicação JEE são sistemas que fornecem a infraestrutura de serviços para a execução de aplicações distribuídas. Simplificando um pouco... servidor de aplicação é um software que fornece um ambiente para a execução das aplicações.

Um servidor de aplicação disponibiliza um ambiente para a instalação e a execução de aplicações. Os servidores de aplicação também são conhecidos como middleware (faz o “meio de campo”). O foco do servidor de aplicações é a resolução de problemas relacionados ao negócio, e não de questões de infraestrutura da aplicação. O servidor de aplicações cuida de algumas questões comuns a todas as aplicações, tais como segurança, garantia de disponibilidade, balanceamento de carga, entre outras.

Servidores de aplicação representam a borda entre a programação e a infraestrutura, por isso o seu estudo é bem abrangente, e incluem os conceitos relacionados a JEE. Seu principal papel é servir a aplicação com serviços de infraestrutura (aquela camada mais abaixo, a base para um bom funcionamento).

Com toda essa base pronta, fica mais fácil o compartilhamento de componentes e aplicações, o desenvolvimento, a manutenção e o gerenciamento de sistemas complexos. Algumas das principais facilidades providas por um servidor de aplicações são:

- Gerenciamento: servidor auxilia a decidir se desligamos a máquina ou pode mantê-la em execução em situações como, por exemplo, a realização de atualizações de software enquanto o site é executado;
- Gerenciamento de transações: lida com problemas como o acesso simultâneo à mesma linha do banco de dados ou quando o banco de dados “cai”;
- Segurança: gerência de permissões de usuários para executar operações, registro de log para auditoria, entre outros;
- Tolerância a falhas: lida com situações em que um servidor “cai” ou um serviço falha, por exemplo, para que o serviço não pare de funcionar;



- Balanceamento de carga: direciona os usuários ao servidor com menos carga. Se um servidor estiver sobrecarregado, outro servidor poderá ser escolhido;
- Failover: se um servidor deixar de funcionar, os clientes poderão ser redirecionados para outros servidores sem a interrupção de serviço. Se isso for possível, é importante que seja o mais rápido possível, e essa definição de tempo aceitável deveria ocorrer de acordo com os requisitos de negócio (1 minuto? 1 hora? ...);
- Mensagens: alguns tipos de solicitações devem ser baseados em mensagem, um sistema em que os clientes e servidores estão muito fracamente acoplados. Portanto, é necessária uma infraestrutura para acomodar um sistema de mensagens;
- *Pool* de recursos: quando um cliente não estiver utilizando um servidor, esses recursos preciosos do servidor poderão ser retornados para um pool, a fim de serem reutilizados quando outros clientes se conectarem. Exemplos de recursos: soquetes (como conexões de banco de dados), objetos que “vivem” dentro do servidor etc.

Para finalizar essa parte, é importante saber que existem diversos exemplos de servidores de aplicação que implementam a especificação JEE, tais como: JBoss, Weblogic, Glassfish, OAS, Geronimo, WebSphere, Netweaver, entre outros.



## QUESTÕES COMENTADAS

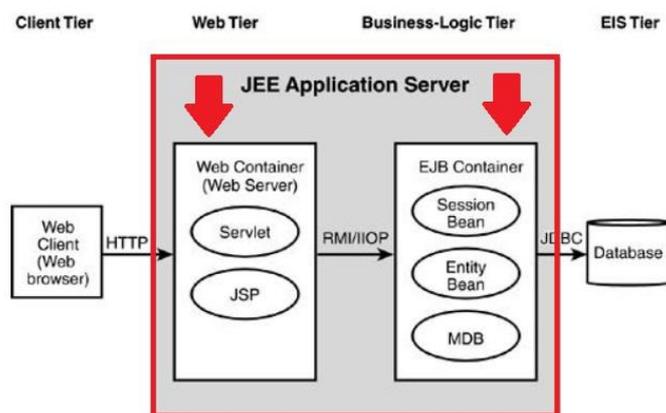
1. (FCC/TRT1 - 2011) J2EE é uma plataforma de programação para servidores na linguagem de programação Java, que integra uma série de especificações e containers, cada uma com funcionalidades distintas.

Nesse contexto, é correto afirmar que são integrantes do J2EE:

- A) Servlets, Jcompany e JSP.
- B) JDBC, JSP, EJBs.
- C) EJBs, Servlets e JBoss.
- D) JDBC, Hibernate e JPA.
- E) JSP, JSF e Eclipse.

### Comentários:

Questão antiga, ainda chamava de J2EE, mas tudo bem. Servlets, JSPs e EJBs integram a especificação JEE. O JBoss é um exemplo de implementação de servidor de aplicação que implementa totalmente a especificação, cuidado! Vamos relembrar a figura abaixo. Podemos ver os servlets e JSP dentro do Web container e os EJBs ficam dentro de um EJB container.



Portanto, a **alternativa B** está correta e é o gabarito da questão.

2. (CESGRANRIO/FINEP - 2011) No contexto da plataforma Java EE (Enterprise Edition), como é chamada a entidade de software que fornece uma interface entre um componente e as funcionalidades de baixo nível específicas da plataforma que suporta tal componente?

- A) Servlet
- B) Class



- C) JSP
- D) Service
- E) Container

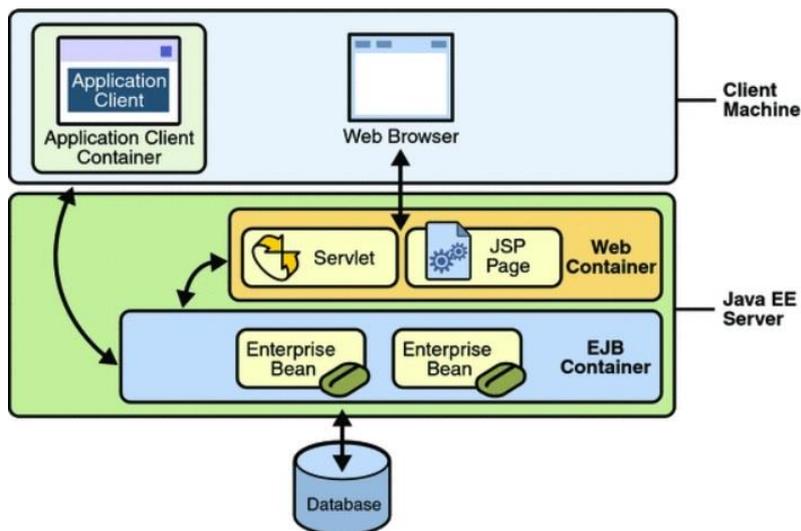
**Comentários:**

Container é um recipiente de objetos Java, ou, de uma maneira mais formal, entidade de software que fornece uma interface entre um componente e as funcionalidades de baixo nível específicas da plataforma Java. Portanto, a **alternativa E** está correta e é o gabarito da questão.

**3. (FCC/TJ-PE - 2012) Sobre JEE e tecnologias relacionadas é correto afirmar que um servidor de aplicações Java EE possui um único contêiner conhecido como contêiner EJB. Além disso, Servlets e JSP rodam no contêiner EJB do servidor de aplicação JEE.**

**Comentários:**

Um servidor de aplicações JEE não possui um único contêiner. Vamos ver outra figura:



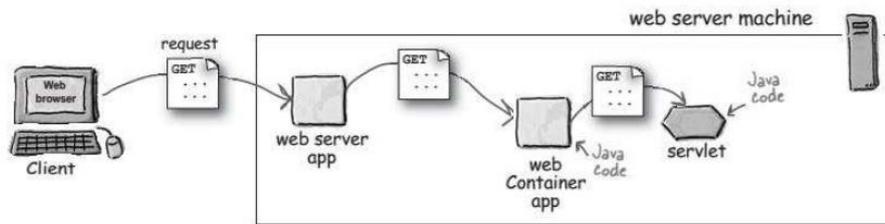
Veja que existem o contêiner Web (contendo Servlets e páginas JSP) e o contêiner EJB (contendo as lógicas de negócio). Portanto, a questão está **errada**.

**4. (FCC/TRE-CE - 2012) No contexto do ciclo de vida de um servlet, quando o servidor recebe uma requisição, ela é repassada para o container que, por sua vez, carrega a classe na memória e cria uma instância da classe do servlet.**

**Comentários:**

Quando uma aplicação Web recebe uma requisição para um servlet, o servidor Web (o Apache, por exemplo) a encaminha não diretamente ao servlet, mas para o container do servlet:





No ciclo de vida dos servlets, quando o servidor recebe uma requisição, ele não a processa diretamente, nem repassa a requisição para o servlet. Na verdade, a requisição é repassada para o container, que as gerencia. Um dos ganhos obtidos com o container é este, ter um objeto que passa a gerenciar o ciclo de vida de um componente. Portanto, a questão está **correta**.

**5. (CESGRANRIO/LIQUIGAS - 2012) Devido à popularização da plataforma Java, o termo "servidor de aplicação" é frequentemente usado como um sinônimo de "servidor de aplicação J2EE". Nesse contexto, os servidores a seguir implementam a especificação J2EE, EXCETO**

- A) WebSphere Application Server
- B) Oracle WebLogic Server
- C) Glassfish
- D) JBoss AS
- E) IIS

**Comentários:**

Questão mais antiga, ainda utilizava o nome J2EE, mas tudo bem. O JBoss é um servidor de aplicação ou servidor de aplicação JEE, ou seja implementa totalmente a especificação JEE. O JBoss, WebSphere, WebLogic e Glassfish também são servidores que implementam a especificação JEE. E o IIS (Information Internet Services)? Bom, esse é um servidor Web da Microsoft! Portanto, a **alternativa E** está correta e é o gabarito da questão.

**6. (IBFC/TRE-AM - 2014) A plataforma JEE (Java Platform, Enterprise Edition) contém uma série de especificações e containers, cada uma com funcionalidades distintas, o Container utilizado no acesso ao banco de dados é conhecido como:**

- A) JSP
- B) JABD
- C) JMS
- D) JDBC

**Comentários:**



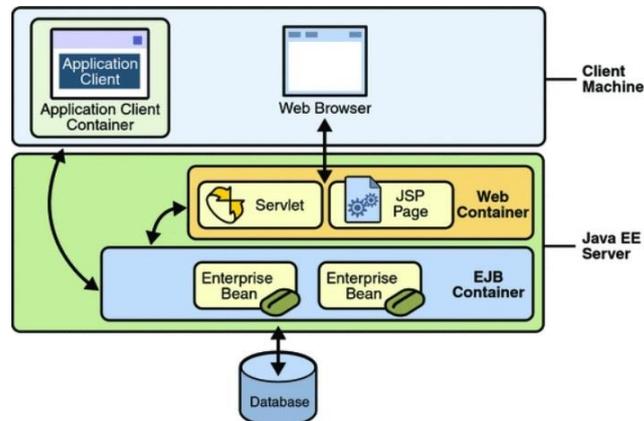
JEE é uma “super especificação” que engloba outras especificações. Entre elas, vimos que o JEE especifica várias APIs, para persistência (JPA), transações (JTA), mensageria (JMS), entre outras. A API que define conexão a bancos de dados é a JDBC. Com esse monte de “letrinhas”, note o DB no meio da sigla, aí fica mais fácil associar com “Data Base” = Banco de Dados. Portanto, a **alternativa D** está correta e é o gabarito da questão.

**7. (FUNCA/MDA - 2014) Um servidor de aplicação JAVA EE pode ser considerado mais completo que um Web Container, pois além de fornecer um Web Container, esse servidor também fornece um:**

- A) Servlet Container.
- B) JAAS Container.
- C) JSP Container.
- D) EJB Container.
- E) JDBC Container.

#### Comentários:

Questão fácil, depois de vermos algumas...não é? Resumindo: Servidor de aplicação JEE = Web container (Servlets, JSP) + EJB container. Para reforçar, aquela figura:



Portanto, a **alternativa D** está correta e é o gabarito da questão.

**8. (FCC/TCM-GO - 2015) Um Analista de Controle Externo da área de TI do TCM/GO está trabalhando em uma aplicação web utilizando a plataforma Java EE. Ciente que os componentes desta aplicação, para serem processados no servidor de aplicações, terão que ser implantados (deployed) em contêineres apropriados, ele esclareceu à equipe de desenvolvimento que servlets, JavaServer Pages, páginas HTML e Java Beans serão implantados e processados no contêiner A. Além disso, alguns componentes serão implantados e processados no contêiner B, destinado a prover a infraestrutura necessária para a execução de componentes de negócio distribuídos que estendem as funcionalidades de um servidor,**



permitindo encapsular lógica de negócio e dados específicos de uma aplicação. Os contêineres A e B são conhecidos, respectivamente, como

- A) local container e remote container.
- B) web container e EJB container.
- C) glassfish container e tomcat container.
- D) EJB container e web container.
- E) server container e client container.

#### Comentários:

Servlets, Java Server Pages (JSP), páginas HTML e Java Beans são implantados e processados no Web container, responsável por gerir o ciclo de vida. O EJB container permite encapsular lógica de negócio e dados específicos de uma aplicação. Portanto, a **alternativa B** está correta e é o gabarito da questão.

**9. (CESPE/ABIN - 2018) Uma das finalidades de um servidor de aplicação web é fornecer facilidades para que o desenvolvedor seja capaz de lidar com a heterogeneidade das especificações de hardware dos servidores.**

#### Comentários:

O servidor de aplicação abstrai modelos e dados oriundos da camada de dados para atender requisições Web. Então, sua finalidade é fornecer facilidades aos desenvolvedores no acesso aos dados de negócio. “Quem” tem que lidar com o hardware é o sistema operacional! Portanto, a questão está **errada**.

**10. (CESPE/ABIN - 2018) Em servidores de aplicação, cache diminui significativamente a carga em banco de dados, especialmente em aplicações que somente realizam leitura no banco, assim, cache em memória é melhor que cache em disco, que é melhor que um banco de dados remoto.**

#### Comentários:

O acesso aos dados em aplicações que realizam muitas consultas (leituras) pode constituir um gargalo. Para reduzir essa possibilidade de gargalo é possível adotar uma estratégia de cache em memória, dos dados mais frequentemente ou recentemente acessados. A cache em memória possui tempos de acesso menores que o acesso em cache em disco ou em bancos de dados remotos. Portanto, a questão está **correta**.

**11. (COSEAC/UFF - 2019) O J2EE é um exemplo de aplicação potencialmente utilizada em:**

- A) VPN.
- B) web services.



- C) PKI.
- D) RSS.
- E) SQL.

#### Comentários:

Mesmo sendo uma questão recente, ainda com o nome antigo! Vimos que o JEE é um servidor de aplicações, sendo um middleware (software que fornece serviços para softwares aplicativos além daqueles disponíveis pelo sistema operacional), ou seja, dando uma base para os aplicativos. Quando se fala em middleware, um padrão bastante utilizado são os Web Services! E para completar, as outras opções são siglas ligadas à segurança (VPN - Virtual Private Network, PKI - Public Key Infrastructure), à sites (RSS - Rich Site Summary) ou acesso a banco de dados (SQL – Structured Query Language). Portanto, a **alternativa B** está correta e é o gabarito da questão.



# SERVIDOR DE APLICAÇÃO JBOSS/WILDFLY

JBoss é um servidor de aplicação de código fonte aberto baseado na plataforma JEE, é implementado completamente na linguagem de programação Java, podendo ser utilizado em qualquer sistema operacional que suporte essa linguagem.

O JBoss Application Server (AS) é a versão *open source*, desenvolvida em comunidades JBoss/RedHat, traz inovações em um ritmo mais rápido com foco em novas funcionalidades, mas seu principal diferencial é que ele não possui suporte oficial.

O JBoss Enterprise Application Platform (EAP) é a versão paga, normalmente evolui a partir das inovações das versões estáveis do JBoss AS, é integrada com recursos como o JBoss Developer Studio e o com o JBoss Operations Network, após um processo de testes (desempenho, escalabilidade etc.), além de possuir suporte oficial.

O JBoss é uma plataforma de middleware<sup>1</sup> baseada em padrões abertos, e que mantém conformidade com a especificação Java EE. É servidor de aplicação que provê recursos de alta disponibilidade, *clustering*, mensageria, cache distribuído, entre outros. Além disso, ele também inclui APIs e frameworks de desenvolvimento para aplicações Java EE escaláveis.

A partir da versão 8 o JBoss passou a se chamar Wildfly, além de ter várias melhorias e mudanças como a troca do container que era o JBossWeb para o Undertow.

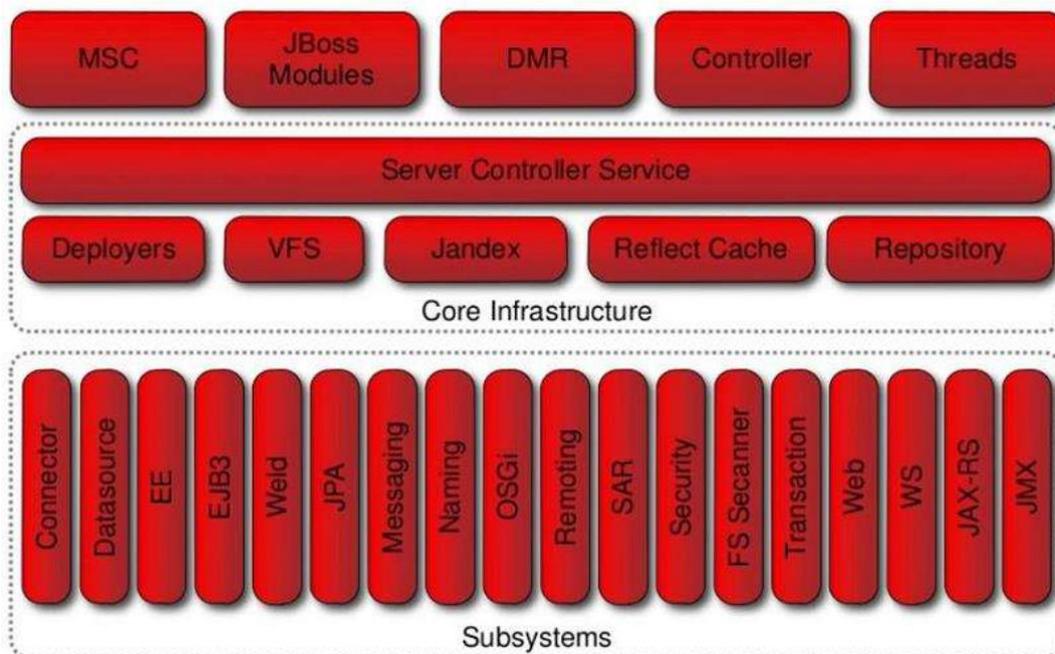
## Arquitetura e Modos

Na figura a seguir podemos ver a arquitetura do JBoss/Wildfly, cabendo um destaque aos subsistemas, onde encontramos algumas API JEE, tais como EJB, JPA, JMX, Mensageria, entre outros.

---

<sup>1</sup> Software que se encontra entre o sistema operacional e os aplicativos nele executados, funcionando de forma essencial como uma camada oculta de tradução. Permite a comunicação e o gerenciamento de dados para aplicativos distribuídos.





Com essa figura fica claro que o JBoss é um servidor de aplicação que implementa diversas características da especificação JEE. Um conceito essencial nas últimas versões é o de domínio. Em um servidor JBoss, um domínio fornece gerenciamento centralizado de várias instâncias de servidor e hosts físicos, enquanto um servidor standalone permite uma única instância do servidor. Configurações, implantações, socket, módulos, extensões e propriedades do sistema podem ser gerenciadas para grupo de servidores. Vamos ver mais detalhes desses dois modos a seguir.

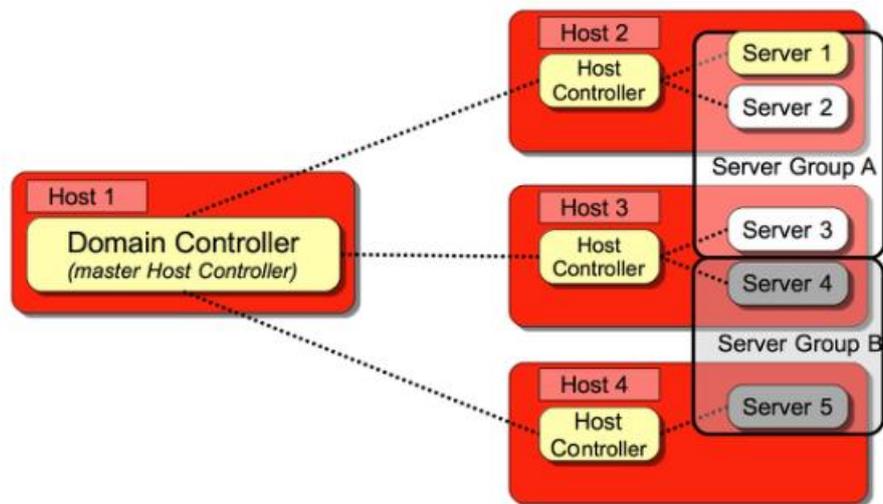
**Modo Standalone:** é o modo tradicional das versões anteriores. Basicamente implica em ter uma instalação diferente (ou um diretório standalone diferente) para cada instância de Wildfly. Ou seja, para cada Wildfly em execução no seu ambiente é necessário alterar seus próprios arquivo de configuração, suas próprias opções de execução para JVM etc. A inicialização nesse modo é realizada ao iniciar o script `JBOSS_HOME/standalone.sh` (no Linux) ou `JBOSS_HOME/standalone.bat` (no Windows).

**Modo Domain:** é o modo que foi introduzido no JBoss AS 7, permitindo gerenciar um conjunto de instâncias Wildfly, agrupando-os e assim permitindo compartilhar configurações comuns entre eles. Além de compartilhar configurações, é possível também através de um único console de gerenciamento iniciar ou parar instâncias (ou grupos inteiros), verificar seu status e estatísticas de cada subsystem etc. O gerenciamento das instâncias é coordenado pelo Domain Controller, tendo várias instâncias (JVMs) por Host e o controle total do ciclo de vida dos servidores via Host Controller. Para iniciar no modo domain é executado o script `JBOSS_HOME/domain.sh` no Linux ou `JBOSS_HOME/domain.bat` no Windows.

Quando o `domain.sh` (ou o `domain.bat`) é executado em um host, um processo conhecido como Host Controller é lançado, o qual é responsável pelo gerenciamento do servidor. Ele não lida diretamente com a carga de aplicações no servidor, sendo responsável por iniciar e terminar os processos que rodam em cada servidor, interagindo com o Domain Controller para ajudar no gerenciamento.

Por padrão, cada Host Controller lê sua configuração no arquivo `domain/configuration/host.xml`, que possui informações de configuração específicas do host. Abaixo podemos ver uma figura com quatro hosts, um Domain Controller e três Host Controllers, sendo que cada Host Controller pode ter um ou mais servidores.





A figura acima mostra uma configuração em modo domain, sendo que todas as configurações e o gerenciamento são realizados de forma centralizada, no Domain Controller. Todas as configurações realizadas para o Domain são replicadas nas instâncias JBoss. Adicionar um novo grupo de servidores, configurar logs, criar data sources, alterar portas, entre outras coisas, tudo isso é feito no [domain.xml](#) do arquivo de configuração do Domain Controller (master).

Características como clustering, high availability (HA), fail-over e outros recursos do JEE estão disponíveis nos dois modos. Domain Controller é quem controla o gerenciamento do domain. Nele estão as configurações que são compartilhadas entre as instâncias que estão nesse domain, e a política de gerenciamento de todos os servidores. O Domain Controller é basicamente um processo Host Controller que dependendo da arquitetura se torna o Domain Controller.

Foi reduzida a necessidade de editar arquivos de configuração XML manualmente. O gerenciamento da segurança é realizado de forma simplificada, inclusive para domínios de segurança. O diretório "modules" centraliza os módulos do servidor de aplicações, em vez do diretório "lib". Os diretórios "domain" e "standalone" possuem os arquivos de configurações para *deploys* (implantações) em modo domain e standalone.

O mecanismo de carga de classe é modular, então os módulos são carregados e descarregados sob demanda, o que propicia uma melhora na segurança, além de menores tempos para iniciar ou reiniciar o servidor. Duas bibliotecas iguais de versões diferentes podem conviver no mesmo servidor.

## Instalação, Configuração e Implantação

Para a instalação, existe a opção de baixar o JBoss/Wildfly em binário, em arquivo zip ou com um instalador. A maneira mais rápida é fazer o download do binário, e descompactar em um diretório. Há alguns riscos que devem ser evitados na instalação, para evitar falhas de segurança. Assim, após a instalação, há a necessidade de ajustar e personalizar a configuração do servidor, antes de colocar o servidor em produção.

O pré-requisito para instalar o JBoss é o Java Development Kit (JDK) ou Java Runtime Environment (JRE) instalado. O download do JBoss/Wildfly pode ser realizado no site da RedHat. Após o download, os arquivos de instalação devem ser colocados na pasta adequada, para iniciar a instalação.



Após a instalação, é necessário configurar o JBoss/Wildfly como serviço. Não é recomendado iniciá-lo, e deixá-lo configurado com o usuário root, pois isto pode comprometer a segurança de toda a plataforma, já que a plataforma Java oferece APIs para execução de códigos nativos do sistema operacional e mecanismos de gerenciamento remoto.

O ideal é que no Linux seja criado um usuário com privilégios adequados para iniciar o serviço do JBoss, e no Windows seja criado um usuário com poderes administrativos, mas com privilégios reduzidos. Como as questões de concurso geralmente abordam comandos (para a configuração ou para a administração), vamos dar uma olhada...

Após a instalação, é necessário criar um grupo:

```
# groupadd jboss
```

Para adicionar o usuário jboss no grupo jboss:

```
# useradd -s /bin/bash -d /home/jboss -m -g jboss jboss
```

Na sequência, a criação da estrutura de diretórios para armazenar o JBoss, atribuindo o dono e o grupo "jboss" a essa estrutura:

```
# mkdir /EAP_HOME/jboss
```

```
# chown jboss:jboss /EAP_HOME/jboss
```

```
# su jboss
```

Depois é necessário configurar a senha para que o usuário utilize na conexão entre o Host Controller e o Domain Controller. Dando continuidade, agora vamos criar os perfis para o domain controller (master) e para os host controllers (slave), baseados no modo Domain e renomeá-los:

```
# cp -Rap /EAP_HOME/domain /EAP_HOME/master
```

```
# cp -Rap /EAP_HOME/domain /EAP_HOME/slave01
```

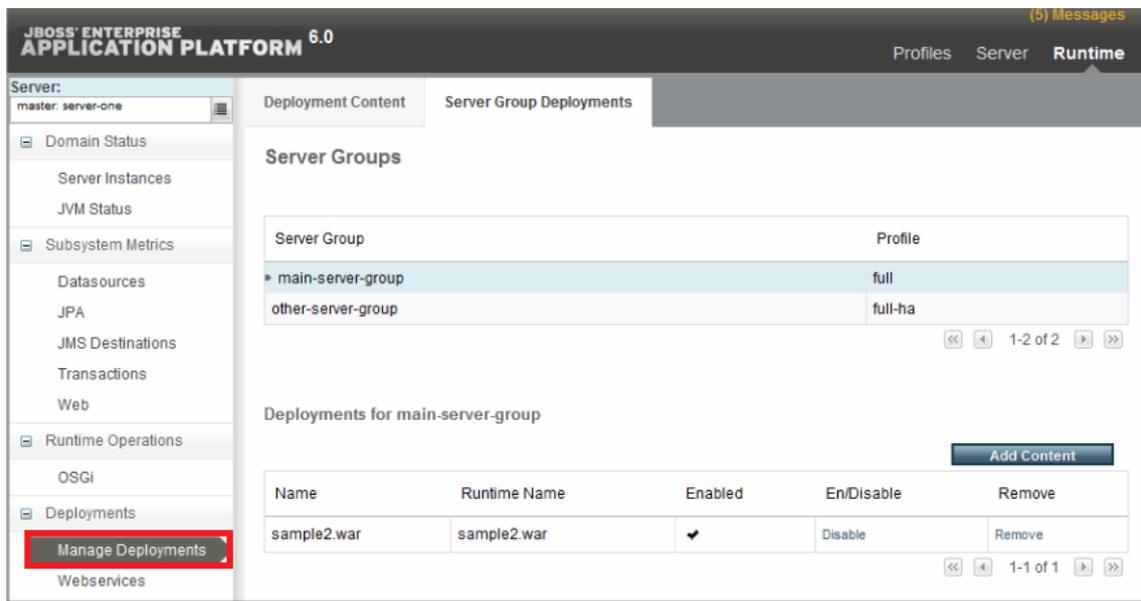
Após a definição do modo (standalone ou domain), é necessário definir um perfil para o JBoss/Wildfly, que é um conjunto de tecnologias ou subsistemas (subsystems) que serão utilizados na aplicação. Essa definição varia conforme os requisitos de cada sistema. **O JBoss possui 4 perfis (profiles) por padrão: default, full, ha, full-ha.**

Depois de concluída a configuração inicial, as aplicações devem ser "colocadas" no JBoss, processo conhecido como *deploy* (implantação).

O *deploy* varia conforme o tipo de perfil (*profile*) definido para o JBoss/Wildfly. Vamos ver como é feito o **deploy de uma aplicação no JBoss EAP**. Primeiro, devemos acessar o console Web na URL <http://x.x.x.x:9990/console/> (onde x.x.x.x é o endereço IP do servidor). Note que a porta utilizada pelo servidor é a 9990, mas roda a aplicação na 8080. Ou seja, o JBoss/Wildfly roda em uma porta e a aplicação em outra.



Em seguida devem ser inseridos o usuário e a senha, aqueles definidos na instalação. Para realizar um *deploy* e o gerenciamento das instâncias pode ser realizado o console:



O *deploy* também pode ser realizado através da linha de comando, o que propicia maior flexibilidade e facilidade na criação de *scripts* de administração. Abaixo um exemplo de *deploy* para todos os servidores:

```
# deploy /path/to/test-application.war --all -server-groups
```

Como já vimos, servidores de aplicação são a interface entre o componente e o sistema operacional específico que o suporta. Antes do componente ser executado no servidor, ele precisa ser montado em uma estrutura que o container possa entender e executar. No JBoss/Wildfly, a **estrutura de empacotamento** é definida na especificação JSR 088 Java EE Application Deployment Specification. **Os formatos padrão são WAR, JAR e EAR.**

## Estrutura de Diretórios

Após a instalação, o JBoss/Wildfly cria uma estrutura de diretórios que deve ser conhecida, pois é necessária para a administração.

- bin/: contém os scripts de inicialização do JBoss EAP, no Linux e no Windows;
- appclient/: contém detalhes de configuração do container da aplicação cliente;
- modules/: contém módulos dinamicamente carregados pelo JBoss EAP quando requeridos pelos serviços;
- standalone/: arquivos de configuração, conteúdo do deploy, e outras áreas utilizáveis quando o JBoss/Wildfly executa como um servidor standalone;
- domain/: contém arquivos de configuração, conteúdo do deploy e outras áreas utilizáveis quando JBoss/Wildfly executa como domínio gerenciado (domain);
- docs/: possui diversos tipos de arquivos como exemplos de configuração, exemplos de como executar o Wildfly como serviço, licenças e outros arquivos que ajudam a aprender mais sobre o Wildfly;
- welcome-content/: diretório de uso interno do servidor que não deve ser modificado por usuários finais, possui páginas de boas vindas e páginas de erros;



- jboss-modules.jar: mecanismo de carga dos módulos.

Além da estrutura de diretórios apresentada, o JBoss/Wildfly cria automaticamente alguns caminhos (*paths*) padrão:

- jboss.home: diretório root do JBoss EAP;
- user.home: diretório de usuário comum;
- user.dir: diretório de trabalho do usuário atual;
- java.home: diretório de instalação do Java;
- jboss.server.base.dir: diretório root de uma instância de um servidor;
- jboss.server.data.dir: diretório que o servidor usa para persistência de dados no storage;
- jboss.server.log.dir: diretório que o servidor usa no armazenamento de logs;
- jboss.server.tmp.dir: diretório de arquivos temporários;
- jboss.domain.servers.dir: diretório no qual um host controller cria a área de trabalho de uma instância em um domínio gerenciado.



## QUESTÕES COMENTADAS

1. (IBFC/EBSERH - 2016) “JBoss é um servidor \_\_\_\_\_ de código fonte \_\_\_\_\_ baseado na plataforma \_\_\_\_\_ e implementado na linguagem de programação \_\_\_\_\_”. Assinale a alternativa que completa correta e respectivamente as lacunas:

- A) de arquivos - aberto - JEE - JavaScript
- B) de aplicação - fechado - JSE - Java
- C) de arquivos - fechado - JEE - JavaScript
- D) de aplicação - aberto - JEE - Java
- E) de impressão - aberto - JSE - JavaScript

### Comentários:

Conceitos básicos que vimos logo no primeiro parágrafo:

JBoss é um servidor de aplicação de código fonte aberto baseado na plataforma JEE, é implementado completamente na linguagem de programação Java, podendo ser utilizado em qualquer sistema operacional que suporte essa linguagem. O JBoss Application Server utiliza o arquivo standalone.bat (ou standalone.sh) para prover a sua inicialização. Portanto, a **alternativa D** está correta e é o gabarito da questão.

2. (FUNCAB/CREA-AC - 2016) Um administrador de rede instalou o Jboss AS 7 no modo domain. Nesse caso, um dos processos principais, que coordena as instâncias e distribui o arquivo implantado para todas as instâncias do domínio, é denominado:

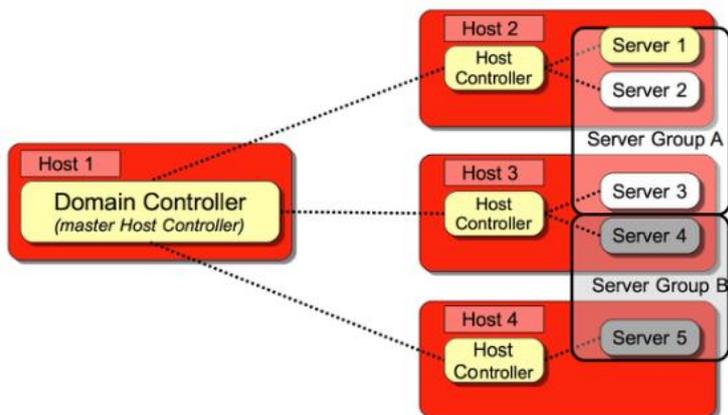
- A) application process.
- B) host controller.
- C) JVM process.
- D) system controller.
- E) standalone controller.

### Comentários:

Quando o domain.sh (ou o domain.bat) é executado em um host, um processo conhecido como Host Controller é lançado, o qual é responsável pelo gerenciamento do servidor. Ele não lida diretamente com a carga de aplicações no servidor, sendo responsável por iniciar e terminar os processos que rodam em cada servidor, interagindo com o Domain Controller para ajudar no gerenciamento.



Por padrão, cada Host Controller lê sua configuração no arquivo domain/configuration/host.xml, que possui informações de configuração específicas do host. Abaixo podemos ver uma figura com quatro hosts, um Domain Controller e três Host Controllers, sendo que cada Host Controller pode ter um ou mais servidores.



Portanto, a **alternativa B** está correta e é o gabarito da questão.

**3. (FCC/TRT11 - 2017) Após instalar o servidor JBoss AS 5 para Windows, deve-se entrar na pasta \$JBOSS\_HOME/bin e digitar run.bat para iniciá-lo. Com o servidor iniciado, para acessar a área de login do JBoss AS Administration Console deve-se digitar,**

- A) na linha de endereço do navegador, `http://localhost:8080/admin-console`.
- B) em linha de comando, `jboss -a console`.
- C) na linha de endereço do navegador, `http://localhost:8084/settings`.
- D) em linha de comando, `jboss -a mode=console`.
- E) na linha de endereço do navegador, `http://localhost:80/server-console`.

#### Comentários:

O deploy varia conforme o tipo de perfil (profile) definido para o JBoss/Wildfly. Vamos ver como é feito o deploy de uma aplicação no JBoss EAP. Primeiro, devemos acessar o console Web na URL `http://x.x.x.x:9990/console/` (onde x.x.x.x é o endereço IP do servidor). Note que a porta utilizada pelo servidor é a 9990, mas roda a aplicação na 8080. Ou seja, o JBoss/Wildfly roda em uma porta e a aplicação em outra.

Note que foi cobrada uma versão mais antiga (AS 5) e teve uma leve variação o nome, mas o que vale é focar na porta 8080, que é a chave da questão!

Portanto, a **alternativa A** está correta e é o gabarito da questão.

**4. (IBFC/EMBASA - 2017) Para que o servidor JBOSS Application Server 7 possa ser plenamente executado, existe a necessidade que esteja previamente instalado e configurado o:**



- A) JMF
- B) JCE
- C) JXL
- D) JDK

#### Comentários:

JBoss é um servidor de aplicação de código fonte aberto baseado na plataforma JEE, é implementado completamente na linguagem de programação Java, podendo ser utilizado em qualquer sistema operacional que suporte essa linguagem.

Para suportar a linguagem Java, deve haver instalado o JDK (Java Development Kit), que é um ambiente utilizado para o desenvolvimento de softwares em Java. O JDK inclui o JRE (Java Runtime Environment), um interpretador/carregador, um compilador (javac), entre outros componentes.

Pode ter apenas o JRE instalado, que também funciona! Mas nas alternativas só encontramos o JDK!

Portanto, a **alternativa D** está correta e é o gabarito da questão.

#### 5. (FGV/Câmara de Salvador-BA - 2018) No âmbito do JBoss AS 7.x, os modos de operação disponíveis são denominados:

- A) Domain e Standalone;
- B) Elevated e Standard;
- C) Local e Remote;
- D) Monouser e Multiuser;
- E) Open e Authenticated.

#### Comentários:

Modo Standalone: é o modo tradicional das versões anteriores. Basicamente implica em ter uma instalação diferente (ou um diretório standalone diferente) para cada instância de Wildfly. Ou seja, para cada Wildfly em execução no seu ambiente é necessário alterar seus próprios arquivo de configuração, suas próprias opções de execução para JVM etc. A inicialização nesse modo é realizada ao iniciar o script JBOSS\_HOME/standalone.sh (no Linux) ou JBOSS\_HOME/standalone.bat (no Windows).

Modo Domain: é o modo que foi introduzido no JBoss AS 7, permitindo gerenciar um conjunto de instâncias Wildfly, agrupando-os e assim permitindo compartilhar configurações comuns entre eles. Além de compartilhar configurações, é possível também através de um único console de gerenciamento iniciar ou parar instâncias (ou grupos inteiros), verificar seu status e estatísticas de cada subsystem etc. O gerenciamento das instâncias é coordenado pelo Domain Controller, tendo várias instâncias (JVMs) por Host



e o controle total do ciclo de vida dos servidores via Host Controller. Para iniciar no modo domain é executado o script `JBOSS_HOME/domain.sh` no Linux ou `JBOSS_HOME/domain.bat` no Windows.

Portanto, a **alternativa A** está correta e é o gabarito da questão.

**6. (FCC/MPE-PE - 2018) Em uma instalação padrão do JBoss Application Server (AS) 7, o diretório que contém a página de boas-vindas do AS é o**

- A) standalone.
- B) welcome-content.
- C) appclient.
- D) server-welcome.
- E) bundles.

**Comentários:**

Após a instalação, o JBoss/Wildfly cria uma estrutura de diretórios que deve ser conhecida, pois é necessária para a administração.

- `bin/`: contém os scripts de inicialização do JBoss EAP, no Linux e no Windows;
- `appclient/`: contém detalhes de configuração do container da aplicação cliente;
- `modules/`: contém módulos dinamicamente carregados pelo JBoss EAP quando requeridos pelos serviços;
- `standalone/`: arquivos de configuração, conteúdo do deploy, e outras áreas utilizáveis quando o JBoss/Wildfly executa como um servidor standalone;
- `domain/`: contém arquivos de configuração, conteúdo do deploy e outras áreas utilizáveis quando JBoss/Wildfly executa como domínio gerenciado (domain);
- `docs/`: possui diversos tipos de arquivos como exemplos de configuração, exemplos se como executar o Wildfly como serviço, licenças e outros arquivos que ajudam a aprender mais sobre o Wildfly;
- `welcome-content/`: diretório de uso interno do servidor que não deve ser modificado por usuários finais, possui páginas de boas vindas e páginas de erros;
- `jboss-modules.jar`: mecanismo de carga dos módulos.

Portanto, a **alternativa B** está correta e é o gabarito da questão.

**7. (FCC/CLDF - 2018) Considere que o servidor de aplicações JBoss AS 7 está instalado e configurado em modo padrão em um computador com sistema operacional Windows 10. Para testar se o servidor JBoss está funcionando, utilizando um navegador, deve-se digitar o URL**

- A) `server://127.0.0.1:0800`
- B) `ftp://localhost:0800`



C) http://localhost:8080

D) server://localhost:80

E) http://127.0.0.1:80

### Comentários:

O deploy varia conforme o tipo de perfil (profile) definido para o JBoss/Wildfly. Vamos ver como é feito o deploy de uma aplicação no JBoss EAP. Primeiro, devemos acessar o console Web na URL `http://x.x.x.x:9990/console/` (onde x.x.x.x é o endereço IP do servidor). Note que a porta utilizada pelo servidor é a 9990, mas roda a aplicação na 8080. Ou seja, o JBoss/Wildfly roda em uma porta e a aplicação em outra. Portanto, a **alternativa C** está correta e é o gabarito da questão.



## LISTA DE QUESTÕES

1. (CESPE/MBASA - 2010) Clusters ou combinações de clusters são usados quando os conteúdos são críticos, apesar de não haver necessidade de estarem disponíveis e (ou) processados rapidamente.
1. (IADES/PG-DF - 2011) Segundo Andrew Tanenbaum (2007) “Sistema Distribuído é uma coleção de computadores independentes que se apresenta ao usuário como um sistema único e consistente”. Assinale a alternativa correta a respeito de um sistema de informação distribuído.
  - A) A distribuição de tarefas se dá a partir de requisições do usuário, que indica o endereço do servidor onde deseja executar tal tarefa.
  - B) Em uma rede de computadores há servidores dedicados a atender pedidos dos clientes e estes, por sua vez, têm função exclusiva de requisitantes.
  - C) Todos os computadores de uma rede executam tarefas de cliente e servidor, quando se deseja integrá-los em uma arquitetura de sistemas distribuídos.
  - D) A transparência de acesso é uma característica dos sistemas distribuídos que permite que recursos sejam acessados sem que sua localização seja determinada.
  - E) Em um sistema de objetos distribuídos é possível invocar métodos de um objeto, ainda que este não esteja presente no computador do usuário.
2. (FCC/TRE-PE - 2011) Arquitetura padrão proposta pelo Object Management Group (OMG) para estabelecer e simplificar a troca de dados entre sistemas distribuídos heterogêneos por meio de uma estrutura comum para o gerenciamento de objetos distribuídos que é conhecida como Object Manager Architecture (OMA). Trata-se de
  - A) IDL.
  - B) RPC.
  - C) DCON.
  - D) CORBA.
  - E) COM.
3. (CESPE/SERPRO - 2013) Com relação à arquitetura de sistemas distribuídos, julgue os próximos itens. Na arquitetura distribuída, os sistemas orientados a eventos possuem processos fortemente acoplados.



**4. (FUNCAB/MDA - 2014) São desvantagens dos sistemas distribuídos:**

- A) dependência de hardware e custo maior de desenvolvimento.
- B) custo maior de desenvolvimento e maior probabilidade de ocorrência de falhas (bugs) no programa.
- C) dependência de sistemas operacionais e necessidade de maior processamento.
- D) necessidade de maior processamento e pouca disponibilidade.
- E) ausência de controle das suas operações e dependência do tipo de rede utilizada para ligar as localidades.

**5. (CESPE/TRE-PE - 2015) Os sistemas operacionais têm rotinas que não são executadas de forma linear, mas, sim, concorrentemente, em função de eventos assíncronos.**

**6. (CESPE/TJ-DFT - 2015) Nos sistemas implementados a partir do uso de uma arquitetura de componentes distribuídos, o middleware tem a responsabilidade de gerenciar a interação entre esses componentes.**

**7. (CESPE/STJ - 2015) A arquitetura orientada a serviços é forma de desenvolvimento de sistemas distribuídos em que os componentes de sistemas são serviços autônomos, razão por que, devido à interoperabilidade, as ligações entre os serviços devem ser rígidas para não provocar mudanças durante sua execução.**

**8. (IBFC/EBSERH - 2017) Assinale a alternativa correta. Cluster é um conceito que está diretamente relacionado aos sistemas de alta disponibilidade. Existem vários tipos de cluster, no entanto há alguns que são mais conhecidos, como:**

- A) cluster paralelo - cluster de alta disponibilidade - cluster de três camadas
- B) cluster de alto desempenho - cluster virtual - cluster matricial
- C) cluster paralelo - cluster virtual - cluster para balanceamento de carga
- D) cluster de alto desempenho - cluster matricial - cluster de três camadas
- E) cluster de alto desempenho - cluster de alta disponibilidade - cluster para balanceamento de carga

**9. (FGV/IBGE - 2017) Keyse é gerente de um centro de dados que hospeda distintos tipos de aplicação. Recentemente Keyse recebeu uma solicitação de hospedagem de uma aplicação de missão crítica, a qual requer uma disponibilidade de 99,8% ao ano.**

**Considerando V para as afirmativas verdadeiras e F para as falsas, analise as características do cluster para atender o requisito de alta disponibilidade do sistema.**



- ( ) capacidade de distribuir igualmente todo o tráfego de entrada entre todos os nós do cluster, para evitar a sobrecarga de requisições a qualquer um deles e o consequente travamento do cluster;
- ( ) ter nós em espera, em quantidade suficiente, para assumir automaticamente a função de outro nó defeituoso;
- ( ) realizar processamento da aplicação de forma paralela entre os vários nós do cluster;
- ( ) monitoramento dos nós feito por eles mesmos, por uma rede diferente da rede de dados.

A sequência correta é:

- A) V – F – V – F;
- B) F – V – F – V;
- C) V – V – V – F;
- D) V – V – F – F;
- E) F – F – V – V.

**10.(CESPE/IPHAN - 2018)** Cluster e GRID possuem características semelhantes, como o multiprocessamento distribuído, mas se distinguem pelas seguintes razões: cluster é uma coleção de máquinas paralelas conectadas entre si que, por trabalharem juntas, fornecem alta disponibilidade e(ou) balanceamento de carga; o GRID, por sua vez, envolve a integração, gestão de recursos computacionais fracamente acoplados e geograficamente distribuídos.

**11.(UFLA/UFLA - 2018)** O aumento da facilidade de acesso à Internet tem permitido uma grande disponibilização da informação. Para dar suporte a essa facilidade de acesso é necessária uma enorme infraestrutura de hardware e software. Considerando as características de computadores paralelos, analise as proposições a seguir:

- I. Um sistema de multiprocessamento simétrico pode ser composto por milhares de computadores com processadores e sistemas operacionais heterogêneos.
- II. Um sistema de processamento paralelo em massa visa resolver problemas que exigem capacidade de utilização de memória compartilhada usando um único conjunto de núcleos de processamento localizados em um mesmo computador.
- III. Um cluster de computadores é uma coleção de dois ou mais computadores usados para executar um dado problema podendo conter processadores multicore.
- IV. Um grid computing provê uma plataforma na qual recursos computacionais são organizados dentro de um ou mais conjuntos lógicos, as tarefas são divididas entre diversos computadores locais ou remotos formando um “super computador virtual”.



**Assinale a alternativa CORRETA:**

- A) Somente as proposições I e II estão corretas.
- B) Somente as proposições II e III estão corretas.
- C) Somente as proposições III e IV estão corretas.
- D) Somente as proposições I e IV estão corretas.

**12.(INSTITUTO AOCP/PC-ES - 2019) Se um servidor Web estiver sobrecarregado, basta agregar novos servidores e dividir a carga de processamento. Considerando o funcionamento dos principais serviços de rede, como é conhecido o esquema de agregação de servidores?**

- A) SIP Apache.
- B) Divserver.
- C) Cluster.
- D) Upload.
- E) ARPANET.

**13.(UFMG/UFMG - 2019) Um Cluster de Alta Disponibilidade é caracterizado por**

- A) no máximo 2 servidores para redundância ou tolerância a falhas e fontes de energia redundantes, dentre outros dispositivos.
- B) diversos servidores para redundância ou tolerância a falhas e fontes de energia redundantes, dentre outros dispositivos.
- C) diversos servidores conectados por uma rede wifi e fontes de energia redundantes, dentre outros dispositivos.
- D) um único servidor com diversas fontes de energia redundantes ligados a um gerador para que esse servidor quase nunca desligue por falta de energia.

**14.(FCC/TRT1 - 2011) J2EE é uma plataforma de programação para servidores na linguagem de programação Java, que integra uma série de especificações e containers, cada uma com funcionalidades distintas.**

**Nesse contexto, é correto afirmar que são integrantes do J2EE:**

- A) Servlets, Jcompany e JSP.
- B) JDBC, JSP, EJBs.



- C) EJBs, Servlets e JBoss.
- D) JDBC, Hibernate e JPA.
- E) JSP, JSF e Eclipse.

**15. (CESGRANRIO/FINEP - 2011) No contexto da plataforma Java EE (Enterprise Edition), como é chamada a entidade de software que fornece uma interface entre um componente e as funcionalidades de baixo nível específicas da plataforma que suporta tal componente?**

- A) Servlet
- B) Class
- C) JSP
- D) Service
- E) Container

**16. (FCC/TJ-PE - 2012) Sobre JEE e tecnologias relacionadas é correto afirmar que um servidor de aplicações Java EE possui um único contêiner conhecido como contêiner EJB. Além disso, Servlets e JSP rodam no contêiner EJB do servidor de aplicação JEE.**

**17. (FCC/TRE-CE - 2012) No contexto do ciclo de vida de um servlet, quando o servidor recebe uma requisição, ela é repassada para o container que, por sua vez, carrega a classe na memória e cria uma instância da classe do servlet.**

**18. (CESGRANRIO/LIQUIGAS - 2012) Devido à popularização da plataforma Java, o termo "servidor de aplicação" é frequentemente usado como um sinônimo de "servidor de aplicação J2EE". Nesse contexto, os servidores a seguir implementam a especificação J2EE, EXCETO**

- A) WebSphere Application Server
- B) Oracle WebLogic Server
- C) Glassfish
- D) JBoss AS
- E) IIS

**19. (IBFC/TRE-AM - 2014) A plataforma JEE (Java Platform, Enterprise Edition) contém uma série de especificações e containers, cada uma com funcionalidades distintas, o Container utilizado no acesso ao banco de dados é conhecido como:**



- A) JSP
- B) JABD
- C) JMS
- D) JDBC

**20. (FUNCA/MDA - 2014) Um servidor de aplicação JAVA EE pode ser considerado mais completo que um Web Container, pois além de fornecer um Web Container, esse servidor também fornece um:**

- A) Servlet Container.
- B) JAAS Container.
- C) JSP Container.
- D) EJB Container.
- E) JDBC Container.

**21. (FCC/TCM-GO - 2015) Um Analista de Controle Externo da área de TI do TCM/GO está trabalhando em uma aplicação web utilizando a plataforma Java EE. Ciente que os componentes desta aplicação, para serem processados no servidor de aplicações, terão que ser implantados (deployed) em contêineres apropriados, ele esclareceu à equipe de desenvolvimento que servlets, JavaServer Pages, páginas HTML e Java Beans serão implantados e processados no contêiner A. Além disso, alguns componentes serão implantados e processados no contêiner B, destinado a prover a infraestrutura necessária para a execução de componentes de negócio distribuídos que estendem as funcionalidades de um servidor, permitindo encapsular lógica de negócio e dados específicos de uma aplicação. Os contêineres A e B são conhecidos, respectivamente, como**

- A) local container e remote container.
- B) web container e EJB container.
- C) glassfish container e tomcat container.
- D) EJB container e web container.
- E) server container e client container.

**22. (CESPE/ABIN - 2018) Uma das finalidades de um servidor de aplicação web é fornecer facilidades para que o desenvolvedor seja capaz de lidar com a heterogeneidade das especificações de hardware dos servidores.**



**23.(CESPE/ABIN - 2018)** Em servidores de aplicação, cache diminui significativamente a carga em banco de dados, especialmente em aplicações que somente realizam leitura no banco, assim, cache em memória é melhor que cache em disco, que é melhor que um banco de dados remoto.

**24.(COSEAC/UFF - 2019)** O J2EE é um exemplo de aplicação potencialmente utilizada em:

- A) VPN.
- B) web services.
- C) PKI.
- D) RSS.
- E) SQL.

**25.(IBFC/EBSERH - 2016)** “JBoss é um servidor \_\_\_\_\_ de código fonte \_\_\_\_\_ baseado na plataforma \_\_\_\_\_ e implementado na linguagem de programação \_\_\_\_\_”. Assinale a alternativa que completa correta e respectivamente as lacunas:

- A) de arquivos - aberto - JEE - JavaScript
- B) de aplicação - fechado - JSE - Java
- C) de arquivos - fechado - JEE - JavaScript
- D) de aplicação - aberto - JEE - Java
- E) de impressão - aberto - JSE - JavaScript

**26.(FUNCAB/CREA-AC - 2016)** Um administrador de rede instalou o Jboss AS 7 no modo domain. Nesse caso, um dos processos principais, que coordena as instâncias e distribui o arquivo implantado para todas as instâncias do domínio, é denominado:

- A) application process.
- B) host controller.
- C) JVM process.
- D) system controller.
- E) standalone controller.



**27.(FCC/TRT11 - 2017) Após instalar o servidor JBoss AS 5 para Windows, deve-se entrar na pasta \$JBOSS\_HOME/bin e digitar run.bat para iniciá-lo. Com o servidor iniciado, para acessar a área de login do JBoss AS Administration Console deve-se digitar,**

- A) na linha de endereço do navegador, `http://localhost:8080/admin-console`.
- B) em linha de comando, `jboss -a console`.
- C) na linha de endereço do navegador, `http://localhost:8084/settings`.
- D) em linha de comando, `jboss -a mode=console`.
- E) na linha de endereço do navegador, `http://localhost:80/server-console`.

**28.(IBFC/EMBASA - 2017) Para que o servidor JBOSS Application Server 7 possa ser plenamente executado, existe a necessidade que esteja previamente instalado e configurado o:**

- A) JMF
- B) JCE
- C) JXL
- D) JDK

**29.(FGV/Câmara de Salvador-BA - 2018) No âmbito do JBoss AS 7.x, os modos de operação disponíveis são denominados:**

- A) Domain e Standalone;
- B) Elevated e Standard;
- C) Local e Remote;
- D) Monouser e Multiuser;
- E) Open e Authenticated.

**30.(FCC/MPE-PE - 2018) Em uma instalação padrão do JBoss Application Server (AS) 7, o diretório que contém a página de boas-vindas do AS é o**

- A) standalone.
- B) welcome-content.
- C) appclient.



D) server-welcome.

E) bundles.

**31. (FCC/CLDF - 2018) Considere que o servidor de aplicações JBoss AS 7 está instalado e configurado em modo padrão em um computador com sistema operacional Windows 10. Para testar se o servidor JBoss está funcionando, utilizando um navegador, deve-se digitar o URL**

A) server://127.0.0.1:0800

B) ftp://localhost:0800

C) http://localhost:8080

D) server://localhost:80

E) http://127.0.0.1:80



# GABARITO

GABARITO



- |           |            |            |
|-----------|------------|------------|
| 1. Errado | 12. C      | 23. Errado |
| 2. E      | 13. C      | 24. Certo  |
| 3. D      | 14. B      | 25. B      |
| 4. Errado | 15. B      | 26. D      |
| 5. B      | 16. E      | 27. B      |
| 6. Certo  | 17. Errado | 28. A      |
| 7. Certo  | 18. Certo  | 29. D      |
| 8. Errado | 19. E      | 30. A      |
| 9. E      | 20. D      | 31. B      |
| 10. B     | 21. D      | 32. C      |
| 11. Certo | 22. B      |            |

▪



# ESSA LEI TODO MUNDO CONHECE: PIRATARIA É CRIME.

Mas é sempre bom revisar o porquê e como você pode ser prejudicado com essa prática.



**1** Professor investe seu tempo para elaborar os cursos e o site os coloca à venda.



**2** Pirata divulga ilicitamente (grupos de rateio), utilizando-se do anonimato, nomes falsos ou laranjas (geralmente o pirata se anuncia como formador de "grupos solidários" de rateio que não visam lucro).



**3** Pirata cria alunos fake praticando falsidade ideológica, comprando cursos do site em nome de pessoas aleatórias (usando nome, CPF, endereço e telefone de terceiros sem autorização).



**4** Pirata compra, muitas vezes, clonando cartões de crédito (por vezes o sistema anti-fraude não consegue identificar o golpe a tempo).



**5** Pirata fere os Termos de Uso, adultera as aulas e retira a identificação dos arquivos PDF (justamente porque a atividade é ilegal e ele não quer que seus fakes sejam identificados).



**6** Pirata revende as aulas protegidas por direitos autorais, praticando concorrência desleal e em flagrante desrespeito à Lei de Direitos Autorais (Lei 9.610/98).



**7** Concurseiro(a) desinformado participa de rateio, achando que nada disso está acontecendo e esperando se tornar servidor público para exigir o cumprimento das leis.



**8** O professor que elaborou o curso não ganha nada, o site não recebe nada, e a pessoa que praticou todos os ilícitos anteriores (pirata) fica com o lucro.



Deixando de lado esse mar de sujeira, aproveitamos para agradecer a todos que adquirem os cursos honestamente e permitem que o site continue existindo.