

Aula 01

*Arquitetura e Sistemas Operacionais p/
TRE-PR (Técnico Judiciário -
Programação de Sistemas) - 2020*

Autor:

**Equipe Informática e TI, Evandro
Dalla Vecchia Pereira**

02 de Março de 2020

Servidores de Aplicação JEE (Java Enterprise Edition)	2
<i>Especificação JEE</i>	2
<i>APIs JEE</i>	3
<i>Servlets</i>	4
<i>EJB (Enterprise Java Beans)</i>	5
<i>JSP (Java Server Page)</i>	5
<i>Container JEE</i>	6
<i>Servidores de Aplicação JEE</i>	7
<i>Questões Comentadas</i>	9
Tomcat	15
<i>Diretórios e arquivos</i>	15
<i>Questões Comentadas</i>	18
Lista de Questões	22
GABARITO	26



PROF. EVANDRO DALLA VECCHIA

Autor do livro "Perícia Digital - Da investigação à análise forense", Mestre em Ciência da Computação (UFRGS), Bacharel em Ciência da Computação (PUCRS), Técnico em Redes de Computadores (Ecom/UFRGS) e em Processamento de Dados (Urcamp). Perito Criminal na área de Perícia Digital desde 2004 no Instituto-Geral de Perícias/RS. Professor de pós-graduação em diversas instituições, nas áreas de Perícia Digital, Perícia Criminal e Auditoria de Sistemas. Lecionou na graduação de 2006 a 2017, nas instituições PUCRS, Unisinos, entre outras. Professor em cursos de formação e aperfeiçoamento de Peritos Criminais, Delegados, Inspetores, Escrivães e Policiais Militares.

Áreas de cursos ministrados pelo professor no Estratégia: Computação Forense, Arquitetura de Computadores e Sistemas Operacionais.

Entre em contato:   profevandrodallavecchia



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 foco nesta aula será a especificação JEE (*Java Enterprise Edition*), além de APIs importantes no contexto dos servidores de aplicação. Então, vamos lá...

ESPECIFICAÇÃO JEE

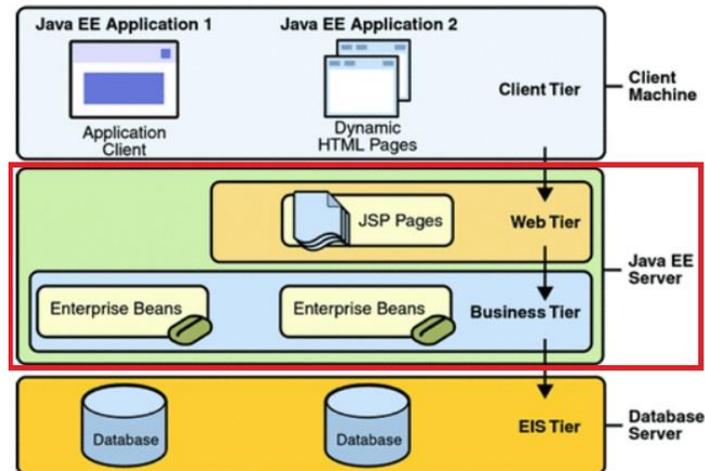
Ao estudarmos a especificação JEE (chamado antigamente de J2EE), veremos alguns conceitos, tais como servlets, Beans, JSP, containers etc., os quais são essenciais para um melhor entendimento do assunto. Mas o que é a JEE? Trata-se de um **conjunto de especificações destinadas a facilitar a criação de 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 transação¹ 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*.

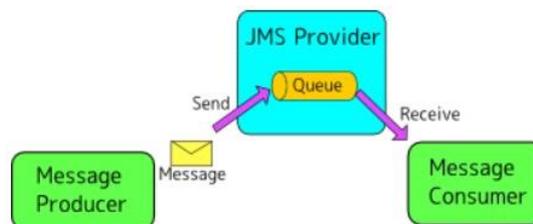
¹ Transação: sequência de operações que são tratadas como um bloco único e indivisível e segue basicamente quatro princípios: atomicidade, consistência, isolamento e durabilidade.

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 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.

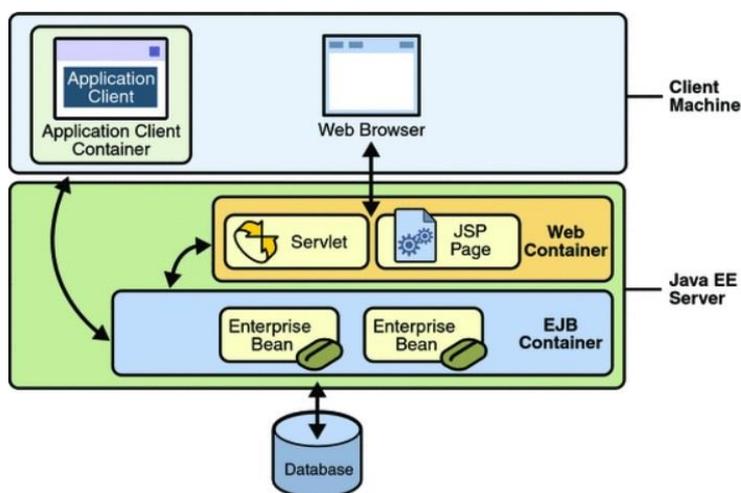
As aplicações baseadas em servlet geram conteúdo dinâmico (geralmente HTML) e interagem com os clientes, utilizando o modelo request/response, normalmente através do protocolo HTTP. Um Servlet necessita de um container Web para ser executado.





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:

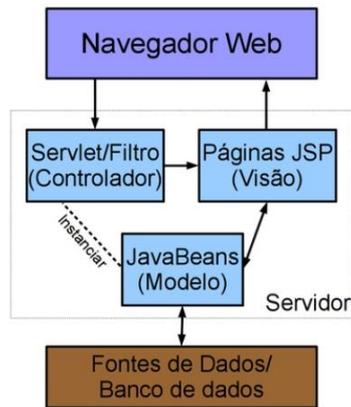


Os principais objetivos da tecnologia EJB são fornecer um rápido e simplificado desenvolvimento de aplicações Java baseado em componentes distribuídos, transacionais, seguros e portáteis.

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.

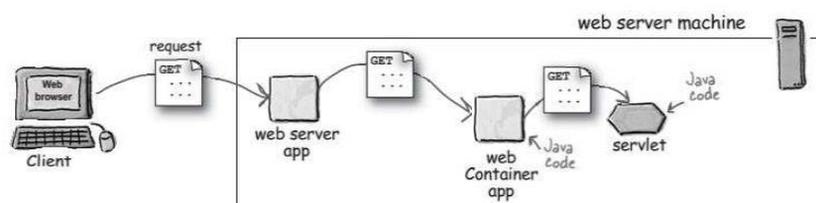
CONTAINER JEE

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.

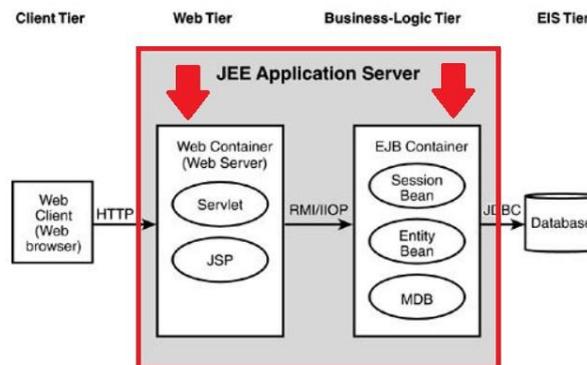
Quando uma aplicação Web recebe uma requisição para um servlet, o servidor Web (o Apache, por exemplo) a encaminha não diretamente ao servlet, mas para o container do servlet:



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. (2011 - FCC - TRT-1ª Região)

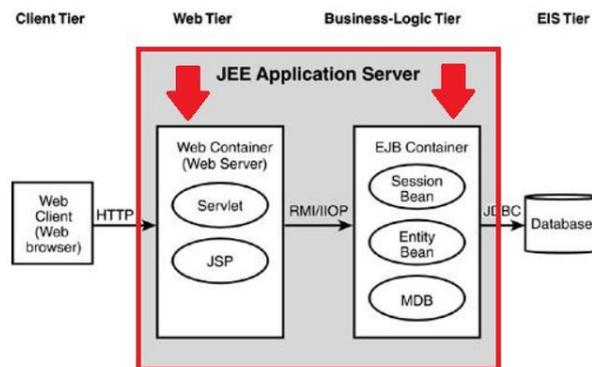
J2EE é uma plataforma de programação para servidores na linguagem de programação Java, que integra uma série de especificações e containers, cada uma com funcionalidades distintas.

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 lembrar a figura abaixo. Podemos ver os servlets e JSP dentro do Web container e os EJBs ficam dentro de um EJB container.



Gabarito: B

2. (2011 - CESGRANRIO - FINEP)

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.

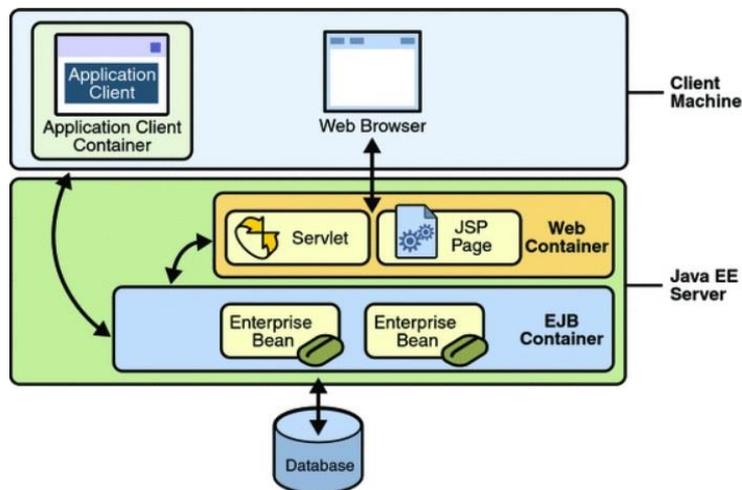
Gabarito: E

3. (2012 - FCC - TJ-PE)

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).

Gabarito: Errado

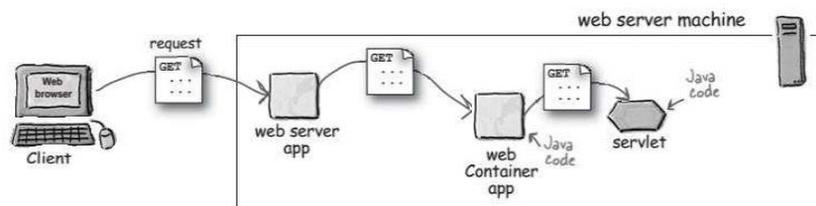
4. (2012 - FCC - TRE-CE)

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.

Gabarito: Certo

5. (2012 - CESGRANRIO - LIQUIGAS)

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!

Gabarito: E

6. (2014 - IBFC - TRE-AM)

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.

Gabarito: D

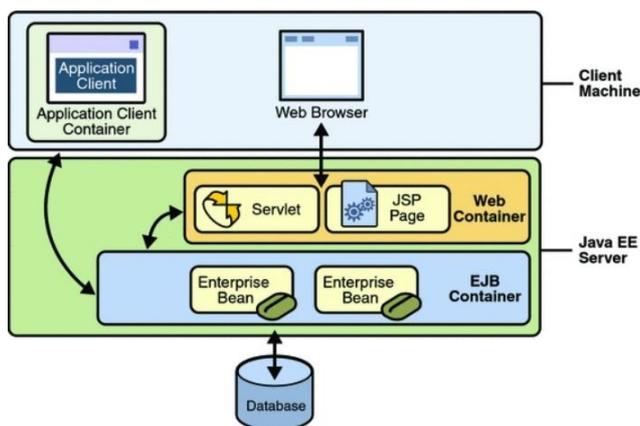
7. (2014 - FUNCA - MDA)

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:



Gabarito: D

8. (2015 - FCC - TCM-GO)

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.

Gabarito: B

9. (2018 - CESPE - ABIN)

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!

Gabarito: Errado

10.(2018 - CESPE - ABIN)

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.

Gabarito: Certo



11.(2019 - COSEAC - UFF)

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*).

Gabarito: B



TOMCAT

O Tomcat é um servidor Web Java, mais especificamente, um container de servlets (por isso é importante ter conhecimento da JEE antes de ler sobre o Tomcat). Trata-se de um software livre com a utilização das tecnologias Servlet, JavaServer Pages (JSPs), Java Expression Language (JEL), Java WebSocket e JASPIC (*Java Authentication Service Provider Interface for Containers*). Importante: não é um container Enterprise JavaBeans (EJB)! Outro detalhe: o Tomcat utiliza por padrão a porta de comunicação 8080.

Quando criado, foi oficialmente endossado pela Sun como a implementação de referência para as tecnologias Servlet e JSP, porém na atualidade a implementação de referência é o GlassFish.

Parte da especificação JEE é coberta (não por completo), tem a capacidade de atuar também como servidor Web, ou pode funcionar integrado a um servidor Web dedicado (ex.: Apache ou IIS). Como servidor Web, ele provê um servidor HTTP puramente em Java. O servidor inclui ferramentas para configuração e gerenciamento, o que também pode ser feito editando-se manualmente arquivos de configuração formatados em XML, e isso é extremamente importante para concursos!

DIRETÓRIOS E ARQUIVOS

Vamos ver alguns diretórios do Tomcat:

- /bin: scripts de inicialização, shutdown, entre outros. Os arquivos ".sh" (sistemas Unix-like) são os equivalentes aos arquivos ".bat" (sistemas Windows);
- /conf: arquivos de configuração DTDs (document type definition) relacionados. O arquivo mais importante nesse diretório é o server.xml, o qual possui a configuração principal para o container;
- /logs: arquivos de log (por padrão);
- /webapps: nesse diretório são colocadas as aplicações Web.

Lendo a documentação do Tomcat, podemos verificar a referência a duas propriedades:

- CATALINA_HOME: representa a raiz da instalação, como por exemplo /home/tomcat/apache-tomcat-9.0.10 ou C:\Arquivos de Programas\apache-tomcat-9.0.10;
- CATALINA_BASE: representa a raiz da configuração de runtime (tempo de execução) de uma instância específica do Tomcat. Se for necessário ter múltiplas instâncias na mesma máquina, deve-se utilizar essa propriedade.

Se as propriedades forem configuradas para diferentes localizações, a CATALINA_HOME contém arquivos estáticos, tais como ".jar" e binários. A CATALINA_BASE possui arquivos de configuração, arquivos de log, aplicações implantadas e outros requerimentos de runtime.



Por padrão, tanto CATALINA_HOME como CATALINA_BASE apontam para o mesmo diretório, então a propriedade CATALINA_BASE deve ser manualmente alterada quando se deseja executar múltiplas instâncias do Tomcat. Com isso, são alcançados alguns benefícios:

- Gerenciamento facilitado para fazer um upgrade para uma nova versão, pois todas as instâncias com uma única localização CATALINA_HOME compartilham um conjunto de arquivos ".jar" e binários, sendo mais fácil fazer o upgrade para a versão mais nova e ter a mudança propagada a todas as instâncias que utilizam o mesmo diretório CATALINA_HOME.
- Evitar a duplicidade dos mesmos arquivos ".jar";
- Possibilidade de compartilhar algumas configurações, como por exemplo um arquivo de script.

Antes de começar a utilizar a CATALINA_BASE, é importante criar a árvore de diretórios utilizada pela CATALINA_BASE. Se não for criada tal árvore, o servidor Tomcat criará tentando automaticamente. Se houver alguma falha (ex.: problemas de permissões), o Tomcat nem conseguirá inicializar, ou pode não funcionar corretamente. Essa é a lista de diretórios:

- "bin" com os arquivos setenv.sh, setenv.bat, e tomcat-juli.jar files;
- "lib" com recursos adicionais serem adicionados no caminho de classes (*classpath*);
- "logs" para arquivos específicos de log;
- "webapps" é carregado automaticamente por aplicações Web;
- "work" contém diretórios temporários de trabalho para as aplicações Web implantadas;
- "temp" é utilizado pela JVM, para arquivos temporários;
- "conf" possui arquivos de configuração.

No mínimo, CATALINA_BASE deve possuir: "conf/server.xml" e "conf/web.xml", senão o servidor Tomcat falha na inicialização ou não funciona apropriadamente. A variável de ambiente CATALINA_BASE pode ser configurada antes de executar o script de inicialização. Exemplos:

- No Unix: CATALINA_BASE=/tmp/tomcat_base1 bin/catalina.sh start
- No Windows: CATALINA_BASE=C:\tomcat_base1 bin/catalina.bat start

Para **gerenciar** um servidor Tomcat e as aplicações remotamente através de uma interface Web, existe a aplicação **Tomcat Manager**. Vamos ver como funciona:

Criando um usuário: no arquivo de configuração que será criado, será configurado um usuário "admin" com a senha "SENHA_SEGURA". Esse usuário terá acesso aos seguintes papéis (*roles*): admin, admin-gui, manager, manager-gui, manager-status, manager-script, manager-xml, admin-script. É possível habilitar somente as roles necessárias para o usuário. Vamos ver um passo a passo:

1. Abrir uma nova janela do "Terminal" ou executar o "Console" em seu painel;
2. Editar o arquivo /var/tomcat/default/conf/tomcat-users.xml, apagar todo o conteúdo do arquivo e, em seguida, colar o conteúdo abaixo (de acordo com a configuração que falamos há pouco):

```
<?xml version='1.0' encoding='utf-8'?>
```



```
<tomcat-users>  
  <role rolename="manager"/>  
  <role rolename="admin"/>  
  <role rolename="admin-script"/>  
  <role rolename="manager-gui"/>  
  <role rolename="manager-script"/>  
  <role rolename="manager-xml"/>  
  <role rolename="manager-status"/>  
  <role rolename="admin-gui"/>
```

```
  <user username="admin" password="SENHA_SEGURA" roles="admin,admin-gui,manager,manager-gui,manager-status,manager-script,manager-xml,admin-script"/>
```

```
</tomcat-users>
```

3. Liberar o acesso de conexão remota ao Tomcat Manager. Para isso, deve-se criar ou editar o arquivo `/var/tomcat/default/conf/Catalina/localhost/manager.xml` e colar o conteúdo a seguir (Obs.: o campo "allow" define quais os endereços terão acesso à conexão remota ao Tomcat Manager. No exemplo abaixo, estão liberados todos os endereços);

```
<Context privileged="true" antiResourceLocking="false" docBase="${catalina.home}/webapps/manager">
```

```
  <Valve className="org.apache.catalina.valves.RemoteAddrValve" allow="^.*$" />
```

```
</Context>
```

4. Liberar o acesso a Role host-manager. Para isso, deve-se alterar o arquivo `/var/tomcat/default/webapps/host-manager/META-INF/context.xml`. Assim como no passo anterior, será necessário definir o campo "allow":

```
<Context antiResourceLocking="false" privileged="true" >
```

```
  <Valve className="org.apache.catalina.valves.RemoteAddrValve" allow="^.*$" />
```

```
</Context>
```

5. Reiniciar o serviço do Tomcat:

```
service tomcat restart
```

6. Pronto! Agora é possível acessar a interface de Administração Web do Tomcat através do endereço `http://SuaConta.dominio.com.br`.



Tomcat Web Application Manager

Message: OK

Manager

List Applications HTML Manager Help Manager Help Server Status

Applications

Path	Version	Display Name	Running	Sessions	Commands
/	None specified	Welcome to Tomcat	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/docs	None specified	Tomcat Documentation	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/examples	None specified	Servlet and JSP Examples	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/host-manager	None specified	Tomcat Host Manager Application	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/manager	None specified	Tomcat Manager Application	true	1	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes

Deploy

Deploy directory or WAR file located on server

Context Path (required):

XML Configuration file URL:

WAR or Directory URL:

Deploy

WAR file to deploy

Select WAR file to upload No file chosen

Deploy

Observação importante: Para que seja possível usar o manager configurado para suportar mais de um host virtual, é necessário que seja realizada a configuração de um manager para cada host!

QUESTÕES COMENTADAS

12.(2016 - INSTITUTO AOCP - EBSERH)

Em sua configuração padrão, o servidor de aplicações Apache Tomcat, utiliza a porta de comunicação

- A) 8010.
- B) 8074.
- C) 8080.
- D) 8686.
- E) 8222.

Comentários:

Um servidor Web “escuta” por padrão as portas 80 (HTTP) e 443 (HTTPS). O servidor de aplicações Tomcat escuta, por padrão, uma porta “parecida” com a de um servidor Web, a 8080.

Gabarito: C



13.(2016 - FAURGS - HCPA)

Qual componente de um servidor Tomcat permite que uma única instância de engine seja vista do exterior, como se fossem múltiplos domínios DNS distintos?

- A) Connector.
- B) Context.
- C) Listener.
- D) Realm.
- E) Virtual host.

Comentários:

Essa questão é sobre o Tomcat, mas veja que puxamos esse conhecimento lá do Apache (servidor Web), o qual possui o Virtual Host para especificar domínios DNS diferentes em uma mesma máquina.

Gabarito: E

14.(2017 - VUNESP - TJM-SP)

No servidor Apache Tomcat 8.5.9, por padrão, a implantação de novas aplicações web (deploy) pode ser realizada copiando o(s) arquivo(s) da aplicação para o diretório:

- A) \$CATALINA_BASE/work
- B) \$CATALINA_BASE/conf
- C) \$CATALINA_BASE/webapps
- D) \$CATALINA_BASE/bin
- E) \$CATALINA_BASE/lib

Comentários:

Essa é a lista de diretórios:

- “bin” com os arquivos setenv.sh, setenv.bat, e tomcat-juli.jar files;
- “lib” com recursos adicionais serem adicionados no caminho de classes (*classpath*);
- “logs” para arquivos específicos de log;
- “webapps” é carregado automaticamente por aplicações Web;
- “work” contém diretórios temporários de trabalho para as aplicações Web implantadas;
- “temp” é utilizado pela JVM, para arquivos temporários;
- “conf” possui arquivos de configuração.

Gabarito: C



15.(2018 - CCV-UFC - UFC)

Uma aplicação nova desenvolvida pela UFC, denominada Ufcnaweb, utiliza a tecnologia Java para web por meio do container de servlets Apache Tomcat. Tal aplicação depende de um arquivo JAR de terceiros denominado writepackage.jar. De acordo com o guia do usuário do Tomcat, onde esse arquivo deverá ser guardado?

- A) Ufcnaweb/WEB-INF
- B) Ufcnaweb/WEB-INF/lib
- C) Ufcnaweb/WEB-INF/jars
- D) Ufcnaweb/WEB-INF/classes
- E) Ufcnaweb/WEB-INF/thirdparty

Comentários:

Note que se trata de um recurso adicional, de terceiros. Essa é a lista de diretórios:

- “bin” com os arquivos setenv.sh, setenv.bat, e tomcat-juli.jar files;
- “lib” com recursos adicionais serem adicionados no caminho de classes (*classpath*);
- “logs” para arquivos específicos de log;
- “webapps” é carregado automaticamente por aplicações Web;
- “work” contém diretórios temporários de trabalho para as aplicações Web implantadas;
- “temp” é utilizado pela JVM, para arquivos temporários;
- “conf” possui arquivos de configuração.

Gabarito: B

16.(2018 - CESPE - STJ)

Acerca de TomCat, desenvolvimento seguro de software, refactoring e integração contínua, julgue o próximo item.

Para que seja possível usar o manager em um Tomcat configurado para suportar mais de um host virtual, é necessário que seja feita a configuração de um manager para cada host.

Comentários:

Bom, o texto está praticamente igual ao que vimos na aula. É isso mesmo, se houver mais de um host virtual, deve haver um manager para cada um.

Gabarito: Certo



17.(2019 - CCV-UFC - UFC)

Sobre o servidor Tomcat 9.0, assinale a alternativa correta.

- A) Utilizando Tomcat 9.0 não é possível a configuração de clusters.
- B) Apache 9.0 funciona como um container Enterprise JavaBeans (EJB).
- C) Tomcat 9.0 implementa a versão Servlet 3.1 e a especificação JSP 2.3.
- D) Tomcat 9.0 implementa toda a especificação Java Enterprise Edition (Java EE).
- E) Tomcat 9.0 implementa várias especificações Java EE, incluindo Java Servlet, JavaServer Pages (JSP), Java EL, WebSocket e especificação JASPIC.

Comentários:

Pode ser um pouco de decoreba, ok...mas é assim mesmo. Vamos retomar nosso primeiro parágrafo desse assunto:

O Tomcat é um servidor Web Java, mais especificamente, um container de servlets (por isso é importante ter conhecimento da JEE antes de ler sobre o Tomcat). Trata-se de um software livre com a utilização das tecnologias Servlet, JavaServer Pages (JSPs), Java Expression Language (JEL), Java WebSocket e JASPIC (*Java Authentication Service Provider Interface for Containers*). Importante: não é um container Enterprise JavaBeans (EJB)! Outro detalhe: o Tomcat utiliza por padrão a porta de comunicação 8080.

Gabarito: E



LISTA DE QUESTÕES

1. (2011 - FCC - TRT-1ª Região)

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.

2. (2011 - CESGRANRIO - FINEP)

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

3. (2012 - FCC - TJ-PE)

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.

4. (2012 - FCC - TRE-CE)

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.



5. (2012 - CESGRANRIO - LIQUIGAS)

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

6. (2014 - IBFC - TRE-AM)

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

7. (2014 - FUNCA - MDA)

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.

8. (2015 - FCC - TCM-GO)

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.

9. (2018 - CESPE - ABIN)

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.

10.(2018 - CESPE - ABIN)

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.

11.(2019 - COSEAC - UFF)

O J2EE é um exemplo de aplicação potencialmente utilizada em:

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

12.(2016 - INSTITUTO AOCP - EBSERH)

Em sua configuração padrão, o servidor de aplicações Apache Tomcat, utiliza a porta de comunicação

- A) 8010.
- B) 8074.
- C) 8080.



- D) 8686.
- E) 8222.

13.(2016 - FAURGS - HCPA)

Qual componente de um servidor Tomcat permite que uma única instância de engine seja vista do exterior, como se fossem múltiplos domínios DNS distintos?

- A) Connector.
- B) Context.
- C) Listener.
- D) Realm.
- E) Virtual host.

14.(2017 - VUNESP - TJM-SP)

No servidor Apache Tomcat 8.5.9, por padrão, a implantação de novas aplicações web (deploy) pode ser realizada copiando o(s) arquivo(s) da aplicação para o diretório:

- A) \$CATALINA_BASE/work
- B) \$CATALINA_BASE/conf
- C) \$CATALINA_BASE/webapps
- D) \$CATALINA_BASE/bin
- E) \$CATALINA_BASE/lib

15.(2018 - CCV-UFC - UFC)

Uma aplicação nova desenvolvida pela UFC, denominada Ufcnaweb, utiliza a tecnologia Java para web por meio do container de servlets Apache Tomcat. Tal aplicação depende de um arquivo JAR de terceiros denominado writepackage.jar. De acordo com o guia do usuário do Tomcat, onde esse arquivo deverá ser guardado?

- A) Ufcnaweb/WEB-INF
- B) Ufcnaweb/WEB-INF/lib
- C) Ufcnaweb/WEB-INF/jars
- D) Ufcnaweb/WEB-INF/classes
- E) Ufcnaweb/WEB-INF/thirdparty



16.(2018 - CESPE - STJ)

Acerca de TomCat, desenvolvimento seguro de software, refactoring e integração contínua, julgue o próximo item.

Para que seja possível usar o manager em um Tomcat configurado para suportar mais de um host virtual, é necessário que seja feita a configuração de um manager para cada host.

17.(2019 - CCV-UFC - UFC)

Sobre o servidor Tomcat 9.0, assinale a alternativa correta.

- A) Utilizando Tomcat 9.0 não é possível a configuração de clusters.
- B) Apache 9.0 funciona como um container Enterprise JavaBeans (EJB).
- C) Tomcat 9.0 implementa a versão Servlet 3.1 e a especificação JSP 2.3.
- D) Tomcat 9.0 implementa toda a especificação Java Enterprise Edition (Java EE).
- E) Tomcat 9.0 implementa várias especificações Java EE, incluindo Java Servlet, JavaServer Pages (JSP), Java EL, WebSocket e especificação JASPIC.

GABARITO

- 1. B
- 2. E
- 3. Errado
- 4. Certo
- 5. E
- 6. D
- 7. D
- 8. B
- 9. Errado
- 10. Certo
- 11. B
- 12. C
- 13. E
- 14. C
- 15. B
- 16. Certo
- 17. E



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.