



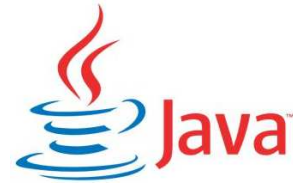
Java Enterprise Edition



Leonardo Marcelino

<http://www.itnerante.com.br/profile/LeonardoMarcelino>

Java Enterprise Edition

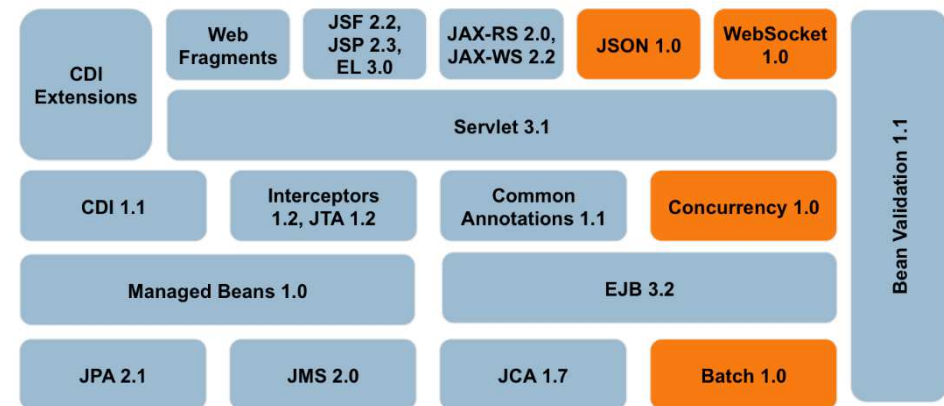
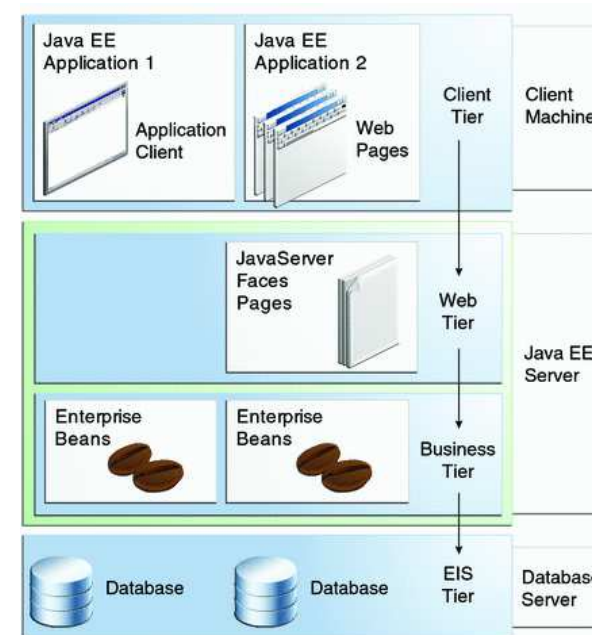


❑ Módulo 1

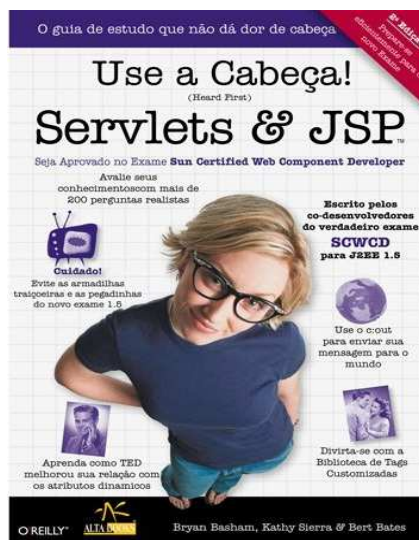
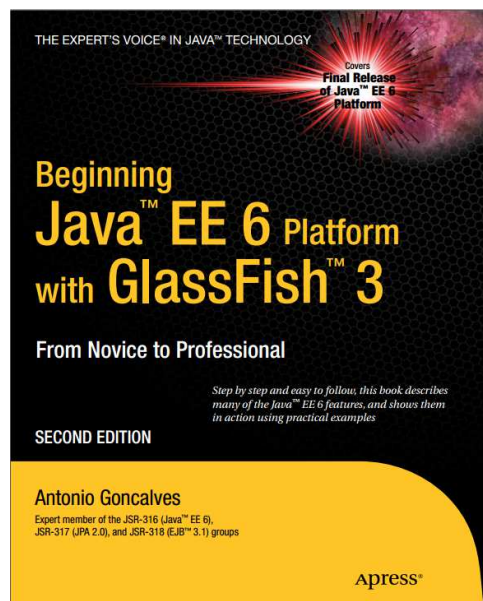
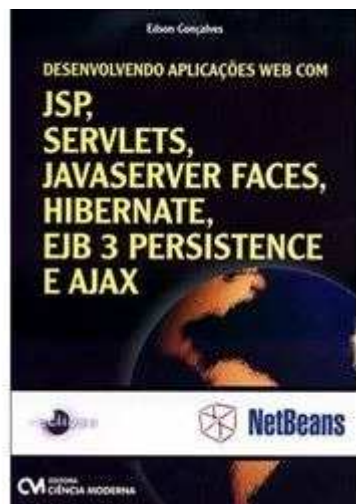
- ✓ linguagem Java
- ✓ conceitos da plataforma JavaEE
 - implementações
 - arquitetura
 - componentes

❑ próximos módulos

- ✓ APIs da plataforma JavaEE
 - Servlets, JSP e JSF
 - EJB
 - outras



REFERÊNCIAS



Linguagem Java

- desenvolvimento de aplicações em diversos ambientes

▪ Java Standard Edition (JSE)

- aplicações desktop com interface gráfica (swing/awt)
- APIs base (JDBC, JNDI, JAAS, JAXP/JAXB)



▪ Java Micro Edition (JME)

- aplicações para dispositivos móveis ou embarcados
- 3 elementos (configuração, perfil e pacotes opcionais)
- configurações: CLDC (simples) e CDC (complexos)

▪ Java Enterprise Edition (JEE)

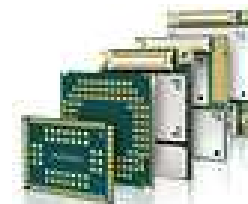
- aplicações corporativas



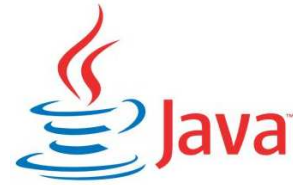
Java Embedded

- sistemas embarcados

- dispositivos de rede, ATMs, equipamentos médicos, etc
- Java SE/ME Embedded, Java DB, Java Card, Java TV, etc



Java Enterprise Edition



❑ conjunto de especificações

- desenvolvimento de aplicações corporativas
- distribuídas (Internet ou Intranet) e server-side

❑ Objetivos da plataforma

- entregar aos desenvolvedores um conjunto de APIs
 - ✓ reduzir o tempo de desenvolvimento
 - ✓ reduzir a complexidade das aplicações
 - ✓ melhorar o desempenho das aplicações
- objetivo mais importante (JEE6)
 - ✓ simplificar o desenvolvimento de aplicações a partir de uma base comum para componentes Java EE



Java Enterprise Edition

☐ melhorias de produtividade (tutorial)

- mais annotations e menos XML (DD)
- mais Plain Old Java Objects (POJOs)
- empacotamento simplificado



☐ Principais Benefícios (faq)



- Produtividade do desenvolvedor
 - ☒ POJOs + Annotations

POJO:

objeto java plano ou regular (desenho simplificado)

não segue modelos, convenções ou frameworks

não estende classes ou implementa interfaces

Annotations (@)

modificador inserido na classe java (metadados)

facilita o deploy evitando configurações em XML

Java Enterprise Edition

❑ Principais Benefícios (faq)

➤ Plataforma rica em recursos

- ☑ APIs padronizadas e integradas



➤ Menor tempo para entrega de soluções ao mercado

- ☑ container gerencia recursos, ciclo de vida e infraestrutura
- ☑ desenvolvedor se preocupa com a lógica do negócio
 - desenvolvedor codifica a lógica nos EJBs
 - container EJB gerencia threads, transações, comunicação, etc



➤ Liberdade de escolha

- ☑ Write Once, Run Anywhere (WORA) no servidor
- ☑ padrões são implementados por vários fornecedores
- ☑ Compatibility Test Suite (CTS) garante compatibilidade/portabilidade
 - Technology Compatibility Kit (TCK)



Java Enterprise Edition

❑ Principais Benefícios (faq)

➤ Suporte completo a Web Services

- ✓ APIs para desenvolvimento/implementação de web services
- ✓ Java API for XML Web Services (JAX-WS) >> SOAP + WSDL
- ✓ Java API for RESTful Web Services (JAX-RS)



➤ Web Profile (JEE6)

- ✓ otimizado para aplicações web
- ✓ não requer a plataforma Java EE completa

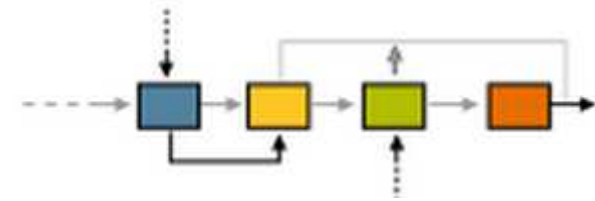


Redução do TCO

- menor tempo de entrega
- liberdade de escolha
- conectividade simplificada



Java Community Process (JCP)



- Java Community
 - qualquer pessoa pode se registrar como JCP membro
 - atuar nas revisões e feedbacks ou participar de Expert Groups
- Comitê Executivo
 - grupo de membros que guiam a evolução de tecnologias Java
 - responsável por aprovar pontos chave no processo (**votação**)
 - selecionar JSRs dentro da JCP e aprovar drafts para revisão pública
 - aprovação final da especificação
- Escritório de Gestão do Programa - PMO (**Oracle**)
 - administra o JCP e preside o Comitê Executivo
- desenvolve especificações de padrões técnicos da tecnologias Java
- garante a estabilidade dos padrões e a compatibilidade multiplataforma da tecnologia Java.
- JSR 355: Java Community Process version 2.9

Java Specification Requests (JSR)

- Documento formal
 - descrições detalhadas
 - requisitos sobre tecnologias da plataforma
 - não é implementação ou framework
 - define padrões para um recurso da plataforma
- Artefatos JSR
 - ✓ - Especificação
 - ✓ - Implementação de Referência (RI)
PoC/protótipo da especificação
 - ✓ - Kit de Compatibilidade de Tecnologia (TCK)
verifica conformidade
implementação x especificação
JCP 1.0: Compatibility Test Suite (CTS)
- Processo de elaboração > JCP 2.9
2.7: Initiation, Early Draft, Public Draft, Maintenance



[1] CESPE 2010 SECGE/PE

J2EE consiste em um padrão para o desenvolvimento de aplicações de negócios que, embora não ofereça um ambiente totalmente seguro e robusto, pode ser escalável por meio de aplicações de controle da organização.

CESPE - 2011 - Correios

O J2EE é um padrão estático que possibilita produzir aplicativos corporativos escaláveis, mas com baixa disponibilidade.

FUMARC - 2011 - BDMG

A plataforma Java EE5 é considerada um padrão de desenvolvimento já que o fornecedor de software nesta plataforma deve seguir determinadas regras se quiser oferecer produtos compatíveis com Java EE5.

[2] FEPESE - 2013 - JUCESC

Sobre as diferenças entre J2EE e Java SE, considere as seguintes afirmativas.

1. J2EE fornece bibliotecas para implementar software Java multicamada e distribuído.
2. Java SE provê suporte a tolerância a falhas enquanto J2EE, não.
3. Os aplicativos J2EE podem ser modulares e fornecidos através de um servidor de aplicação.

Assinale a alternativa que indica todas as afirmativas **corretas**

- | | |
|---------------------------------|---------------------------------|
| a) apenas a afirmativas 2. | b) apenas as afirmativas 1 e 2 |
| c) apenas as afirmativas 1 e 3. | d) apenas as afirmativas 2 e 3. |
| e) as afirmativas 1, 2 e 3. | |

[3] CESPE - 2013 - MPOG

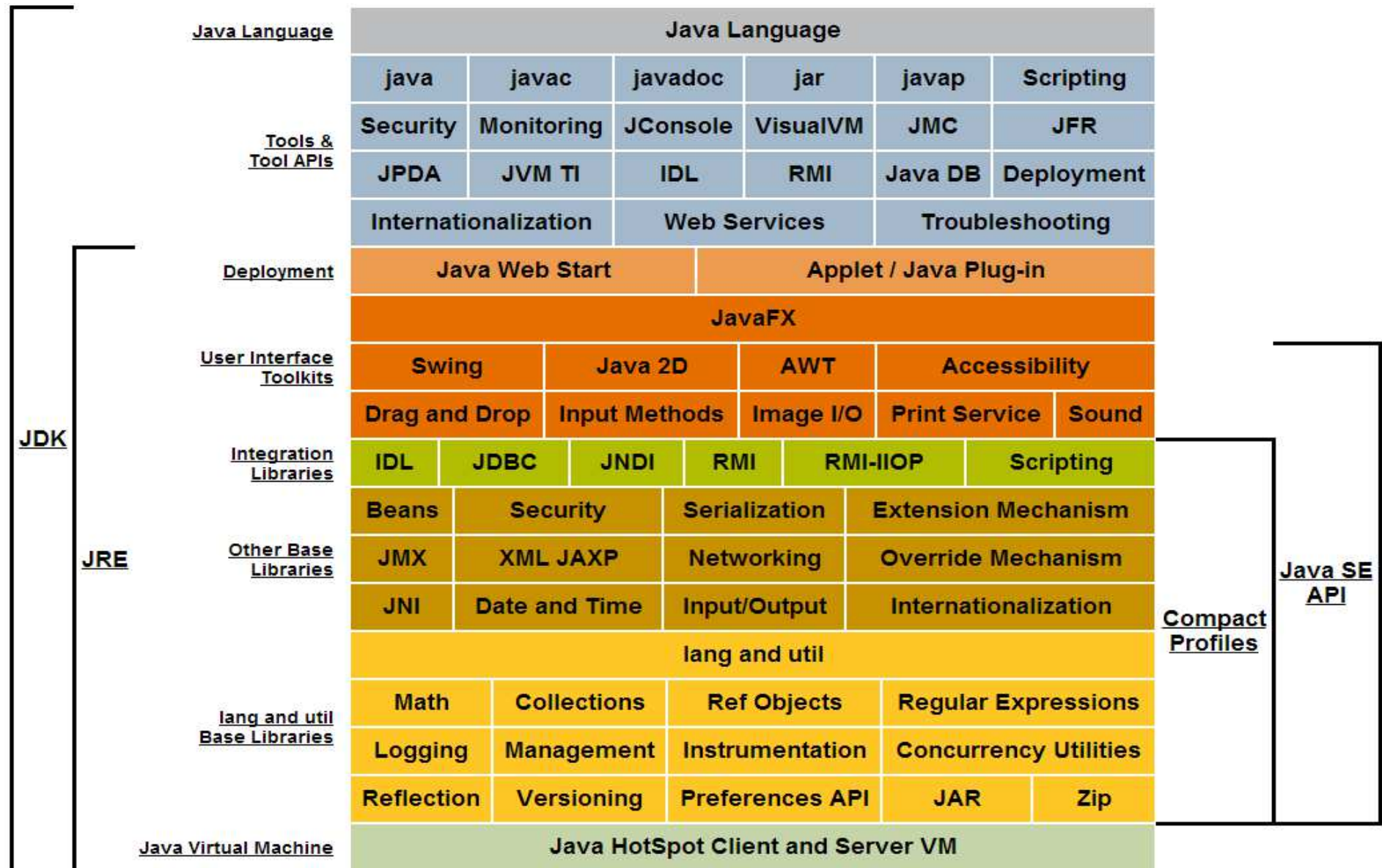
Com referência às tecnologias Java SE (JSE), Java ME (JME) e Java EE (JEE), julgue os itens subsecutivos.

O JME foi criado para trabalhar com ambientes de programação multiprocessados em ambientes distribuídos.

Diferentemente do JSE, o JEE adiciona bibliotecas que proveem funcionalidades para desenvolvimento de aplicações com tolerância a falhas, distribuídas em multicamadas e baseadas em componentes modulares em servidores de aplicação.

O JSE consiste em uma máquina virtual utilizada para rodar programas Java. Para utilizar a interface gráfica, a versão deve ser alterada para JSEG.

Linguagem de programação Java



Implementações

Empresas com produtos que implementam especificações JEE

ORACLE®



Implementação de referência JEE:



servidores de aplicação (lato sensu)

Servidores de aplicação

- Software que implementam especificações JEE
- Permite que aplicações corporativas sejam executadas
 - Infraestrutura para hospedar/tratar componentes
- Abstrair complexidades de baixo nível
 - oferece serviços de **segurança**, controle de **transação**, balanceamento de carga, tolerância a falhas, etc
 - desenvolvedor preocupa com a lógica do negócio
- Exemplos: **JBoss** (RedHat), **WebSphere** (IBM), **WebLogic** (Oracle), **Geronimo** (Apache), **Glassfish** (implementação de referência), etc.

☐ Apache Tomcat 7

servidor de aplicação (lato sensu)

não é servidor de aplicações JEE (strictu sensu) “full”

container para **servlets** e **JSP**

container web (**não tem suporte nativo a JSF**)

Servlet 3.0, JSP/EL 2.2 e WebSocket 1.0



Servidores de aplicação

Java EE 5		
- Apache Geronimo-2.1.4	- JBoss Application Server 5.0	- TmaxSoft JEUS 6
- Oracle WebLogic Server 10g R3	- GlassFish Application Server v2	- OW2 JOnAS 5.1
- IBM WebSphere Application Server v7		

Java EE 6 Full Platform		
- IBM WebSphere Application Server 8.x	- Apache Geronimo 3.0-beta-1	- Oracle GlassFish Server 3.x
- IBM WebSphere Application Server Community Edition 3.0	- Oracle WebLogic Server 12.1.1	- TMAX JEUS 7
	- JBoss Application Server 7.x	
Java EE 6 Web Profile		
- Oracle GlassFish Server 3.x	- Apache TomEE 1.0	
- IBM WebSphere Application Server 8.5.5 (Liberty Profile)	- JBoss Application Server 7.x	
	- JOnAS 5.3	

Java EE 7 Full Platform	Java EE 7 Web Profile
- GlassFish Server Open Source Edition 4.0	- GlassFish Server Open Source Edition 4.0 Web Profile
- TMAX JEUS 8	

JavaEE Profiles

- Servidor de aplicação JEE é muito pesado
 - possui muitos componentes (API) desnecessários
 - aplicações menores não utilizam todo o potencial
- JEE 6 inseriu na plataforma a ideia de perfil (JEE Profile)
 - configuração da plataforma adequada a determinado tipo de aplicação
 - pode conter:
 - ⇒ subconjunto de tecnologias JEE
 - ⇒ tecnologias não JEE (portlet JSR-286)
- **JavaEE Web Profile**
 - conjunto de especificações essenciais para aplicação web
 - primeiro perfil JEE

Apache **Tomcat** não é web profile

✓ TomEE ⇒ JEE 6 Web Profile

Tomcat + MyFaces + OpenEJB



JavaEE Web Profile

JavaEE 6	JavaEE 7
Servlet 3.0	Servlet 3.1
JavaServer Pages (JSP) 2.2	JavaServer Pages (JSP) 2.2
Expression Language (EL) 2.2	Expression Language (EL) 3.0
Debugging Support for Other Languages (JSR-45) 1.0	Debugging Support for Other Languages (JSR-45) 1.0
Standard Tag Library for JavaServer Pages (JSTL) 1.2	Standard Tag Library for JavaServer Pages (JSTL) 1.2
JavaServer Faces (JSF) 2.0	JavaServer Faces (JSF) 2.2
Common Annotations (JSR-250) 1.1	Common Annotations (JSR-250) 1.1
Enterprise JavaBeans (EJB) 3.1 Lite	Enterprise JavaBeans (EJB) 3.2 Lite
Java Transaction API (JTA) 1.1	Java Transaction API (JTA) 1.2
Java Persistence API (JPA) 2.0	Java Persistence API (JPA) 2.1
Bean Validation 1.0	Bean Validation 1.1
Managed Beans 1.0	Managed Beans 1.0
Interceptors 1.1	Interceptors 1.1
Contexts and Dependency Injection 1.0	Contexts and Dependency Injection 1.1
Dependency Injection for Java 1.0	Dependency Injection for Java 1.0
	Java API for RESTful Web Services (JAX-RS) 2.0

✓ requer apenas módulos .WAR

Apache TomEE



	Tomcat	TomEE	OpenEJB
Java Servlets	✓	✓	
Java ServerPages (JSP)	✓	✓	
Java ServerFaces (JSF)		✓	
Java Transaction API (JTA)		✓	✓
Java Persistence API (JPA)		✓	✓
Java Contexts and Dependency Injection (CDI)		✓	✓
Java Authentication and Authorization Service (JAAS)		✓	✓
Java Authorization Contract for Containers (JACC)		✓	✓
JavaMail API		✓	✓
Bean Validation		✓	✓
Enterprise JavaBeans		✓	✓
Java API for RESTful Web Services (JAX-RS)			✓
Java API for XML Web Services (JAX-WS)			✓
Java EE Connector Architecture			✓
Java Messaging Service (JMS)			✓

CDI - Apache OpenWebBeans	JSTL - Apache Tomcat
EJB - Apache OpenEJB	JTA - Apache Geronimo Transaction
JPA - Apache OpenJPA	Servlet - Apache Tomcat
JSF - Apache MyFaces	Javamail - Apache Geronimo JavaMail
JSP - Apache Tomcat	Bean Validation - Apache BVal

[4] CESPE - 2011 - PREVIC

O servidor de aplicações disponibiliza uma plataforma que abstrai do desenvolvedor de *software* algumas das complexidades de um sistema computacional, atendendo, assim, algumas questões comuns a todas as aplicações, como, por exemplo, segurança, garantia de disponibilidade, balanceamento de carga e tratamento de exceções.

[5] CESPE - 2010 - TCU

Um servidor de aplicação JEE age como uma máquina virtual estendida que é voltada para a execução de aplicações e que trata de forma transparente as conexões com o banco de dados e com o cliente.

A diminuição da segurança, a perda de desempenho e o aumento do TCO são alguns dos impactos do emprego de servidores de aplicação JEE.

Uma vantagem do uso de servidores de aplicação JEE é a descentralização da configuração.

[6] CESPE - 2011 - TRE ES

O Tomcat, servidor de aplicações Java para Web, é um *software* livre de código aberto, que surgiu do conceituado projeto Apache Jakarta, com o apoio e endosso oficial da Sun Microsystems como implementação de referência para as tecnologias Java Servlet e JavaServer Pages (JSP).

CESPE - 2011 - TJ-ES

Na qualidade de servidor de aplicações Java para *web*, o Tomcat é parte da plataforma J2EE, que abrange as tecnologias Servlet e JSP, incluindo as tecnologias de apoio relacionadas, tais como Realms e segurança, JNDI Resources e JDBC DataSources. O Tomcat tem a capacidade de atuar também como servidor *web*/HTTP ou pode funcionar integrado a um servidor *web* dedicado, tais como o Apache httpd ou o Microsoft IIS.

[7] FCC - 2011 - TRT 4 (RS)

Considere:

- I. Cobre parte da especificação JEE porém não as tecnologias *servlet* e JSP e nem as tecnologias *Realms* e JDBC *DataSources*.
- II. Tem a capacidade de atuar como servidor web, ou pode funcionar integrado a um servidor web dedicado como o Apache ou até mesmo o IIS.
- III. É distribuído pela Sun como software proprietário e é referência para as tecnologias *Java Servlet* e *JavaServer Pages*.

Quanto ao *Tomcat*, é correto o que consta em

- a) I, somente.
- b) II, somente.
- c) III, somente.
- d) I e II, somente.
- e) I, II e III.

[8] CESPE - 2011 - TJ ES

O Tomcat e o Glassfish, servidores de aplicações Java (*container servlet*), são *open source* e compatíveis com aplicações escritas na arquitetura JEE 6; no entanto, ao contrário do Tomcat, o Glassfish não suporta executar aplicações escritas em JSP (*Java server pages*).

CESPE - 2010 - BASA [adaptada]

O servidor de aplicação JBOSS, em suas versões mais atuais, foi preparado para suportar aplicações que utilizam as especificações *servlets* e JSP, mas são incompatíveis com a especificação EJB.

[9] FCC - 2012 - TRT 11 (AM)

Sobre o Tomcat é correto afirmar:

- a) É um servidor de aplicação que fornece o ambiente necessário para a execução de aplicações Web desenvolvidas com as tecnologias Java Servlet e JSP.
- b) É um software livre (mas de código fechado) para Windows, desenvolvido no projeto Apache Jakarta, que atualmente é definido como a Implementação de Referência (RI) para as tecnologias Java Servlet e JSP.
- c) O Tomcat Manager não permite ao administrador visualizar as aplicações web instaladas, entretanto, permite verificar o status do servidor, instalar, desinstalar, iniciar, parar e recarregar uma aplicação web.
- d) O Tomcat é o nome popular do servidor Web Apache HTTP Server que possui uma variedade de ferramentas capazes de lidar com qualquer tipo de solicitação na Web. Interpreta códigos desenvolvidos em diversas linguagens, como PHP, Perl, Shell Script, ASP etc.
- e) O Tomcat foi escrito em Java e, portanto, necessita de uma Java Virtual Machine (JVM) para ser executado.

[10] CESPE - 2010 - TCU

- O sistema deverá ser embasado na plataforma JEE (Java *enterprise edition*) v.6, envolvendo *servlets*, JSP (Java *server pages*), Ajax, JSF (Java *server faces*) 2.0, Hibernate 3.5, SOA e *web services*.

Com relação às diferentes arquiteturas e tecnologias que, se escolhidas, impactarão as características do sistema descrito no texto, julgue o item.

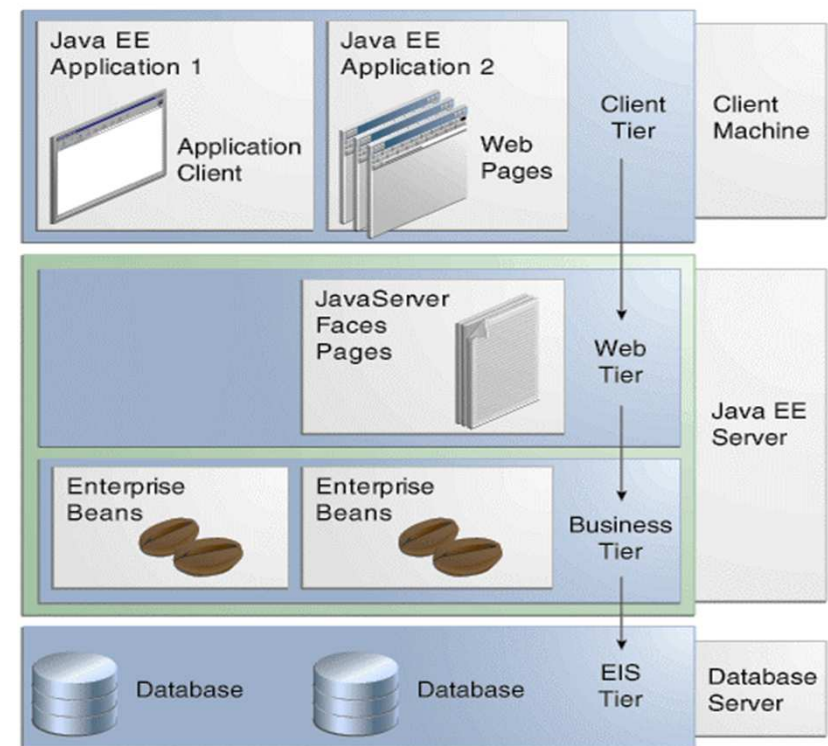
A web profile da plataforma JEE apresenta, em relação ao perfil application server definido em edições anteriores da plataforma Java, as seguintes vantagens: fornece suporte para POJOs (plain old Java objects) e annotations; possui modelo de empacotamento de componentes mais simples; a configuração dos seus descritores XML (extensible markup language) é mais fácil; é aderente ao padrão SOA.

Arquitetura de aplicações JEE

- Modelo para aplicação JEE
 - possui dois componentes básicos:
 - linguagem JAVA (JSE, JEE)
 - máquina virtual Java (JRE)
 - fornece
 - portabilidade, segurança e produtividade
 - define uma arquitetura **multicamadas**
 - características
 - escalabilidade, acessibilidade e gerenciamento
 - divide o workflow em duas perspectivas
 - ⇒ lógica do negócio e apresentação (**desenvolvedor**)
 - ⇒ serviços padrões do sistema (**plataforma JEE**)
 - abstrai complexidades inerentes a aplicações corporativas
 - gerenciamento de transações, ciclo de vida de componentes, recursos, etc.

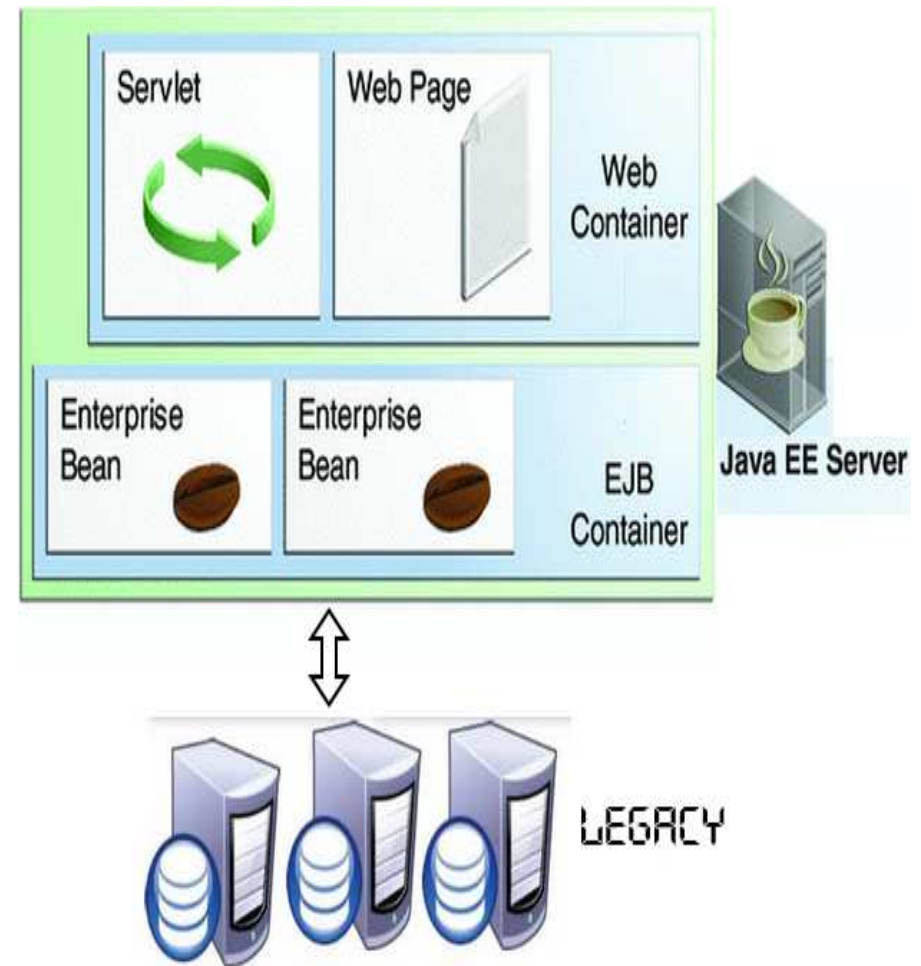
Arquitetura de aplicações JEE

- Arquitetura multicamadas e distribuída
 - constituída de diversos componentes
 - agrupados de acordo com a função que exercem (camadas)
 - instalados em diferentes máquinas (distribuídos)
 - camadas
 - localização (hardware)
 - considerada três camadas
 - estende modelo cliente-servidor
 - servidor de aplicação multithread
 - ⇒ cliente x armazenamento
 - camadas de componentes
 - Cliente
 - Web
 - Negócio
 - Enterprise Information System



Arquitetura de aplicações JEE

- Perspectiva J2EE
 - partes fundamentais de aplicações corporativas
 - Componentes
 - Containers
 - Conectores
 - foco no legado
 - interoperabilidade legado x J2EE
 - conectores
 - externos à plataforma J2EE
 - API Java Connector Architecture
- Perspectiva JEE
 - foco na aplicação JEE
 - produtividade

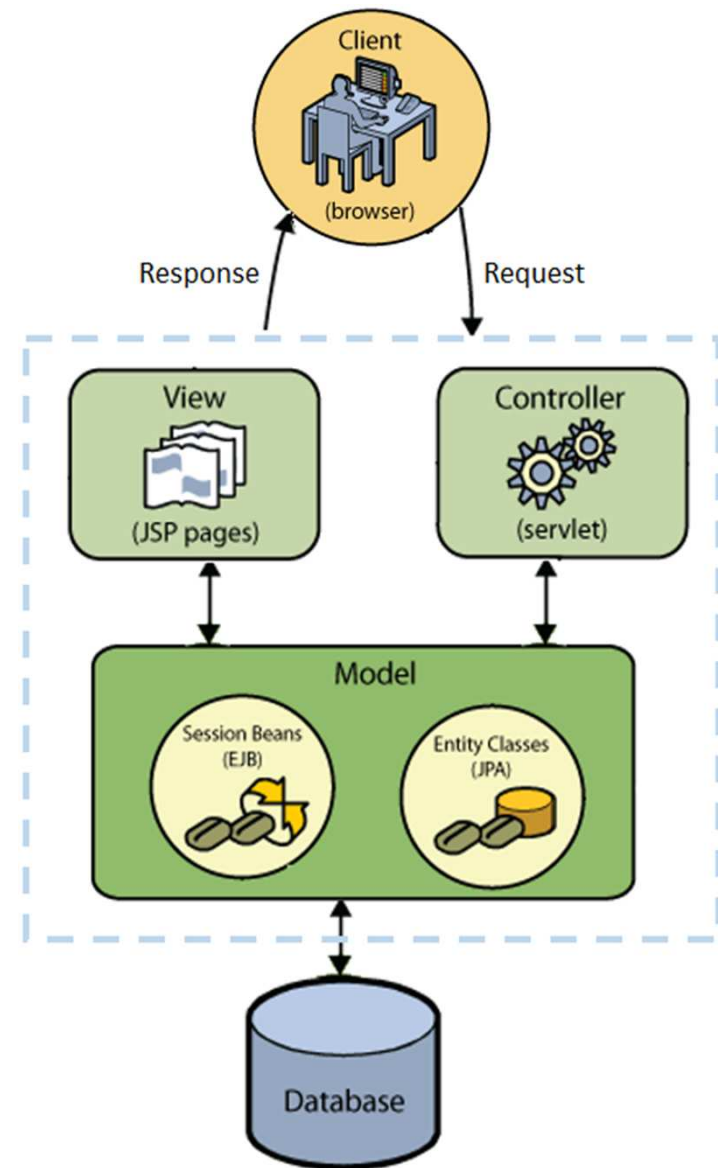


Arquitetura de aplicações JEE

■ Perspectiva MVC

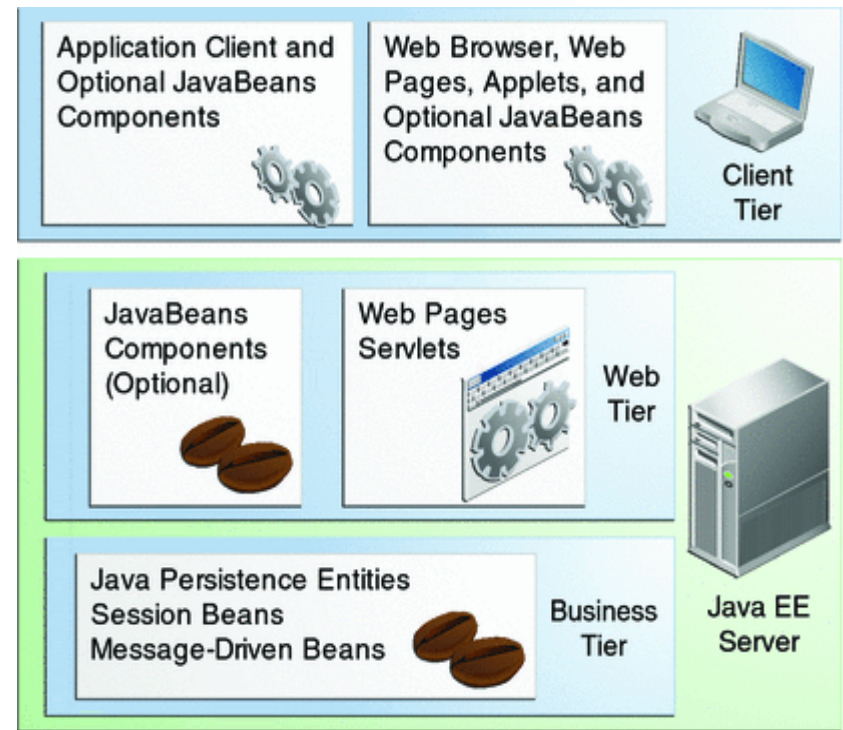
■ MVC

- Padrão arquitetural
- composto de três papéis (partes)
 - Modelo
 - ✓ regras de negócio, estado da aplicação, persistência
 - Visão
 - ✓ renderiza dados do modelo para o usuário
 - Controlador
 - ✓ comportamento da aplicação, intermediário entre modelo/visão
- evitar o termo “camadas”
 - ✓ comunicação em pilha
 - ✓ papéis ou partes



Componentes Java EE

- Aplicação JEE é componentizada
- **Componente JEE?**
 - unidade de software autocontida e funcional
 - conjunto de classes + arquivos
 - compliance com especificações Java EE
- aplicação = componentes (assemble) + interação
- 3 categorias:
 - componentes cliente
 - aplicações clientes e applets
 - componentes web (servidor)
 - Servlets, JSP e JSF
 - componentes de negócio (servidor)
 - EJB
 - **JavaBeans não é componente JEE**



Componentes Java EE

■ Componentes Web

- rodam na camada web
- Servlets e Páginas web (JSF ou JSP)

■ Servlets

- modelo requisição-resposta

■ JavaServer Pages

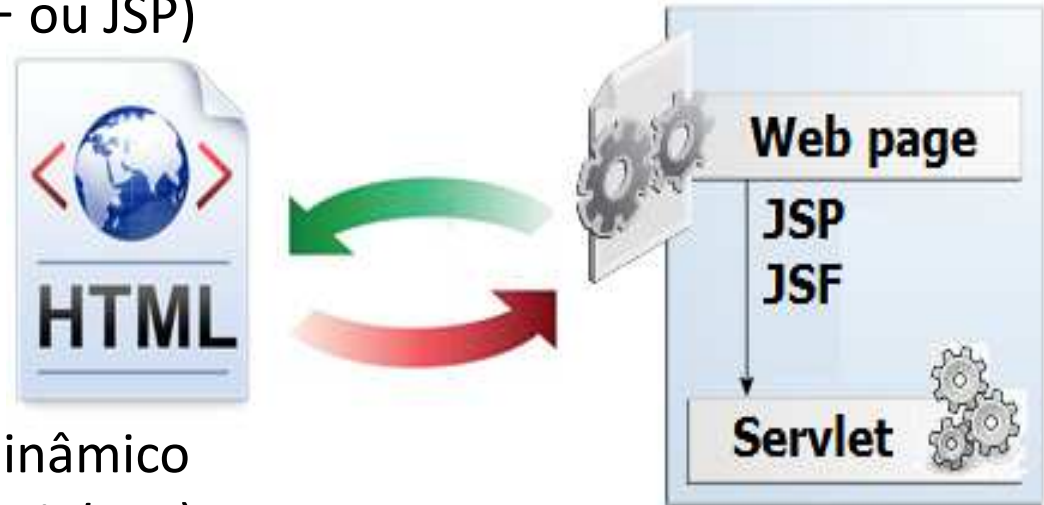
- executado como servlet
- combina código estático + dinâmico
- gera conteúdo estático (*tutorial JEE*)

■ JavaServer Faces

- baseada em JSP + Servlet
- framework de componentes UI

■ Não são componentes web

- applets, páginas estáticas HTML e classes utilitárias



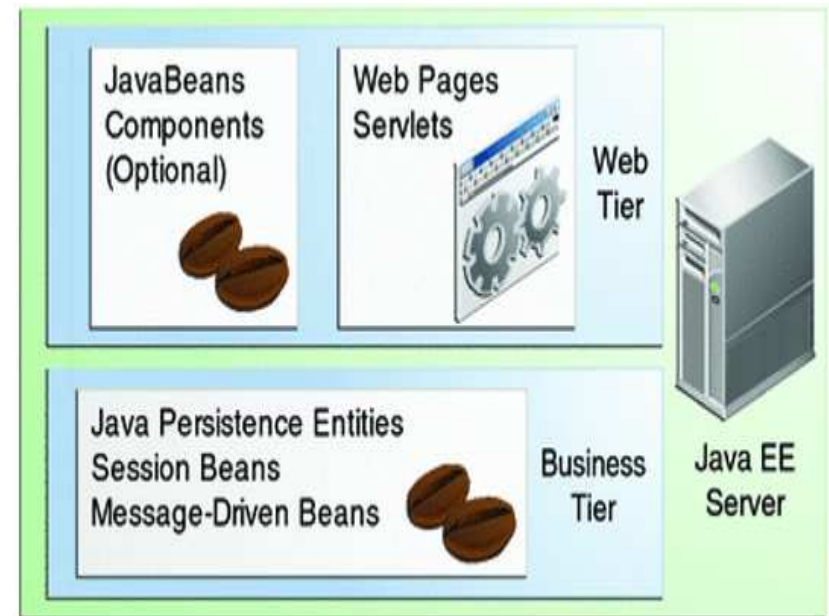
Componentes Java EE

■ Componentes de negócio

- Enterprise Java Beans
- rodam na camada de negócio
 - na camada web (opcional)
- processam a lógica do negócio
 - serviços bancários, e-business, etc.

■ Enterprise Java Bean (EJB)

- interação entre as camadas cliente e EIS
- camada de negócios
 - persistência * (J2EE)
 - controle de sessão
 - troca de mensagens assíncronas
- camada web
 - EJB lite (JEE6)
 - **JavaBeans** (não é componente JEE)
 - acesso a dados



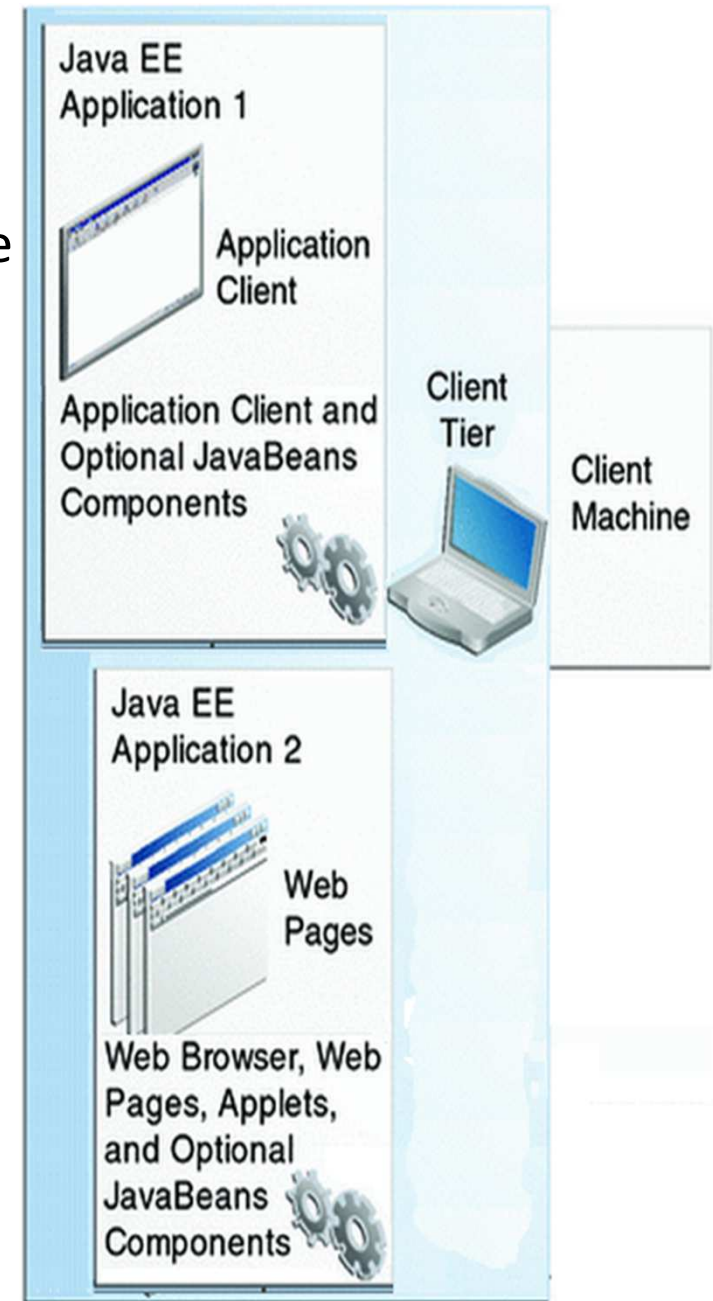
Clientes Java EE

■ Aplicação cliente

- aplicação java instalada / rodando no cliente
 - ✓ requer JVM
- usada para interfaces mais sofisticadas
 - ✓ tarefas/interfaces mais complexas
 - ✓ GUI com Swing ou AWT
- comunica/interage com servidores JEE
 - ✓ EJB na camada de negócio
 - ✓ servlets na camada web (conexão HTTP)
- aplicações não-Java
 - ✓ podem interagir com servidores JEE
 - ✓ interoperabilidade com sistemas legados

■ Applet

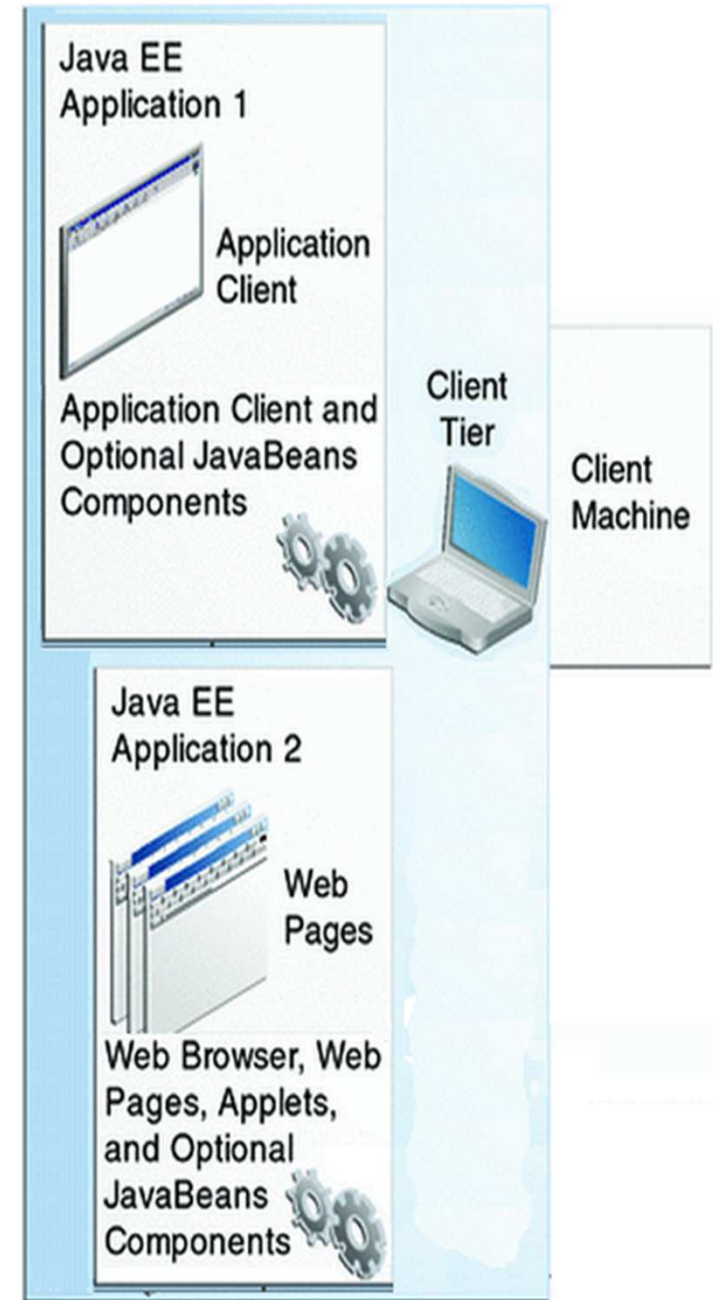
- aplicação Java que roda na JVM do browser
- requer no cliente:
 - ✓ plug-in Java e política de segurança



Clientes Java EE

▪ Cliente web

- cliente magro (thin client)
- não requer instalação no cliente
 - ✓ políticas de segurança no servidor
- roda nos navegadores
- composição
 - ✓ página web (HTML, XML)
 - geradas por componentes da camada web
 - ✓ web browser
 - renderização
- requisição / resposta
 - ✓ sem consultas a SGBD / Legados
 - ✓ regras de negócio complexas no servidor
- distinção entre
 - apresentação (design da página web)
 - das regras de negócio



[11] FCC - 2006 - BACEN

Sobre servidores e aplicações J2EE:

I. A plataforma J2EE utiliza um modelo de aplicação distribuída multicamada;

II. A lógica das aplicações J2EE é dividida em componentes de acordo com a função;

III. Os vários componentes que constituem uma aplicação J2EE são instalados em diferentes equipamentos.

É correto o que consta em

- | | | |
|-----------------------|---------------------|----------------------|
| a) I, II e III. | b) I e II, somente. | c) I e III, somente. |
| d) II e III, somente. | e) III, somente. | |

[12] CESPE - 2013 - CPRM

Julgue os itens subsecutivos a respeito de arquitetura, padrões de projeto e desenvolvimento *web*.

No JEE7, para uma aplicação em três camadas, é correto afirmar que, na camada cliente, seriam distribuídos os componentes JavaServer Pages, JavaServer Faces e componentes JavaBeans; na camada servidor Web, os componentes Enterprise Beans e Java Servlets; e na camada de banco de dados, os componentes SGBD e arquivos XML.

[13] CESPE - 2011 - Correios

Acerca dos fundamentos, características e topologias típicas em ambientes com alta disponibilidade e escalabilidade e da arquitetura J2EE, julgue os próximos itens.

No desenvolvimento de programas em múltiplas camadas, uma camada pode referir-se tanto a uma máquina diferente quanto a uma camada lógica, de modo que múltiplas camadas podem estar em uma mesma máquina. A arquitetura de aplicativos de n camadas físicas, como a fornecida pelo J2EE, envolve a separação da funcionalidade com a utilização de camadas lógicas e físicas e não permite o uso de componentes dentro das camadas lógicas.

[14] ESAF - 2012 - CGU

Os níveis da plataforma J2EE são:

- a) Patrocinador. Web. Negócios. Sistemas de Computação corporativos.
- b) Cliente. Web. Negócios. Sistemas de Informação Corporativos.
- c) Cliente. Interno. Externo. Negócios.
- d) Fornecedor. Web. Político. Sistemas de Informação Camada.
- e) Cliente. *Stakeholders*. Negócios. *Background* corporativo.

[15] CESGRANRIO - 2008 - TJ RO

Considere uma aplicação J2EE que utiliza a arquitetura MVC. A correspondência mais provável entre cada tipo de componente ativo da aplicação com o significado das letras da sigla MVC, nesta ordem, é:

- a) EJB, JSP, Servlet
- b) EJB, Servlet, JSP
- c) Servlet, JSP, EJB
- d) JSP, EJB, Servlet
- e) JSP, Servlet, EJB

[16] CESPE - 2008 - STJ

Acerca de arquiteturas de aplicações cliente-servidor e para a Internet, julgue os próximos itens.

Na plataforma J2EE, uma aplicação web para a Internet pode ser composta por servlets, Java Server Pages (JSP) e páginas HTML. Nessas aplicações, a apresentação dos dados pode ser separada da lógica do negócio, adotando-se o estilo de arquitetura model view controller (MVC). Nesse caso, pode-se usar servlets operando como controladoras que recebem as solicitações dos usuários e providenciam o processamento das mesmas. Em uma mesma aplicação, entretanto, só pode existir um servlet operando como controladora.

[17] FCC - 2011 - TRT 19 (AL)

A especificação Java EE define os seguintes componentes:

- I. Clientes da aplicação (*Application Clients*) e *applets*.
- II. *Java Servlet*, *JavaServer Faces* e *JavaServer Pages*.
- III. *Enterprise Javabeans* (EJB).

Os componentes I, II e III rodam, respectivamente, em:

- a) cliente, cliente, servidor.
- b) servidor, cliente, servidor.
- c) cliente, servidor, servidor.
- d) servidor, cliente, cliente.
- e) cliente, servidor, cliente.

[18] CESPE - 2008 - TRT 5 (BA)

Julgue o item que se segue com relação à arquitetura J2EE.

Applets são componentes que podem ser executados tanto do lado servidor quanto do lado cliente em qualquer aplicação Java.

Enterprise JavaBeans são componentes de negócios em Java que são executados do lado do cliente.

Java Servlets são componentes Java executados somente do lado do servidor.

[19] CESPE - 2008 - TRT 5 (BA)

Com relação a linguagens de programa, julgue os itens que se seguem.

Um cliente J2EE pode ser um cliente *web* ou uma aplicação cliente. A aplicação cliente é formada por dois elementos: páginas *web* dinâmicas, que contêm vários tipos de linguagens de *markup*, geradas por componentes *web*; e um *web browser*, que apresenta as páginas recebidas do servidor.

[20] CESPE - 2010 - TRE MT

Quanto às tecnologias Java, assinale a opção correta.

- a) Clientes J2EE são necessariamente páginas web dinâmicas que normalmente não fazem acessos a banco de dados, nem executam regras de negócio complexas.
- b) Em uma aplicação J2EE típica, um EJB é criado, controlado e destruído pela aplicação cliente.
- c) JSP pode ser usado para desenvolvimento de páginas web dinâmicas. A lógica que gera o conteúdo dessas páginas é descrita utilizando-se a linguagem Java.
- d) Um componente J2EE é uma unidade funcional de software autocontida, escrito na linguagem de programação Java e executado exclusivamente em servidores.
- e) JPA lida com a forma como dados relacionais são mapeados para objetos Java e com a forma como esses objetos são armazenados em um banco de dados relacional.

Containers Java EE

- interface entre componentes e a funcionalidade de baixo-nível
- camadas JEE são implementadas nos containers
- componentes
 - ✓ montados (assembled) em um módulo JEE
 - ✓ implantados (deploy) no container
- serviços **configuráveis** (*deployment descriptors*)
 - ✓ segurança
 - ✓ gerenciamento de transações
 - ✓ serviço de diretório (pesquisas com JNDI)
 - ✓ conectividade remota
 - ⇒ comunicação de baixo nível cliente x componentes

Containers Java EE

- serviços **não-configuráveis**
 - ✓ gerenciamento de ciclo de vida de componentes (EJB, servlet, etc)
 - ✓ acesso a APIs JEE
 - ✓ pooling de recursos (componentes, bancos de dados, etc)
 - ✓ persistência de dados
 - pode desativar o gerenciador de persistência
- thread-safe
 - *single-thread* [EJB], *multi-thread* (**tutorial**)

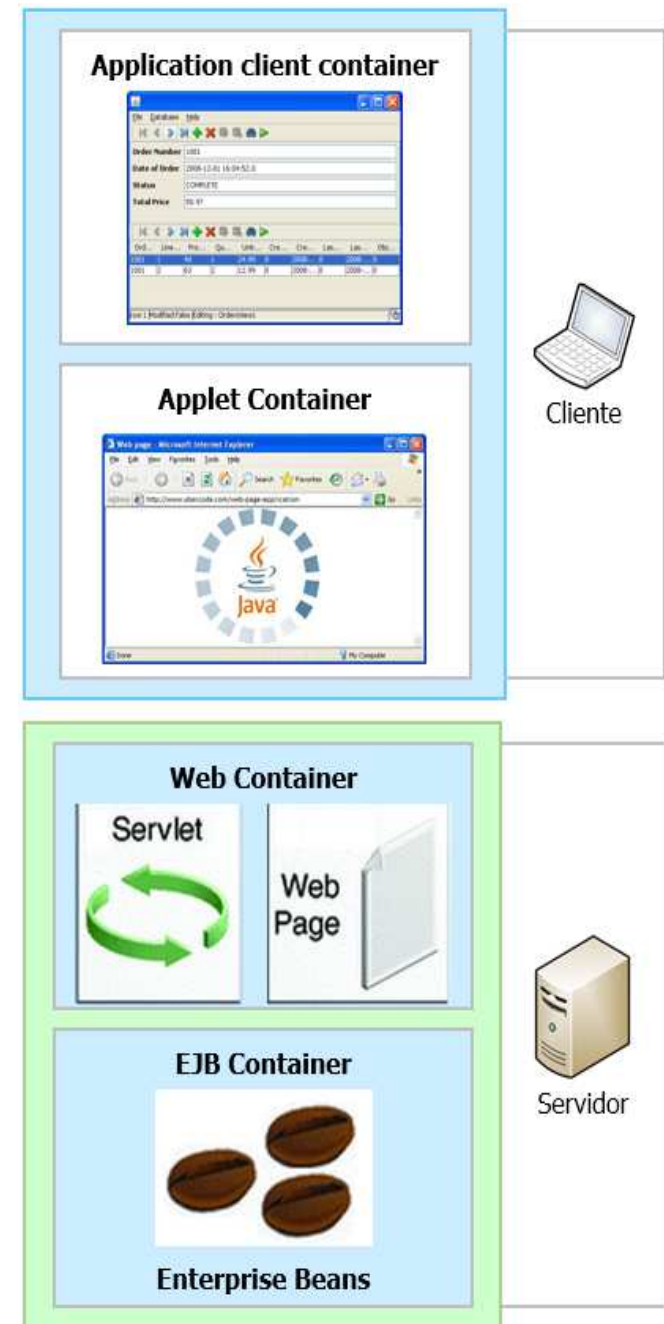
Tipos de Containers Java EE

■ Aplicação cliente

- roda no cliente
- trata componentes específicos da aplicação
 - ✓ conjunto de classes e bibliotecas
- executado na JVM (JRE)
- comunica com o servidor JEE (RMI/IIOP ou HTTP)
- composto de
 - ✓ JavaSE (JRE)
 - ✓ JavaEE
 - persistência, mensageria, webservices e injeção de dependência

■ Applet

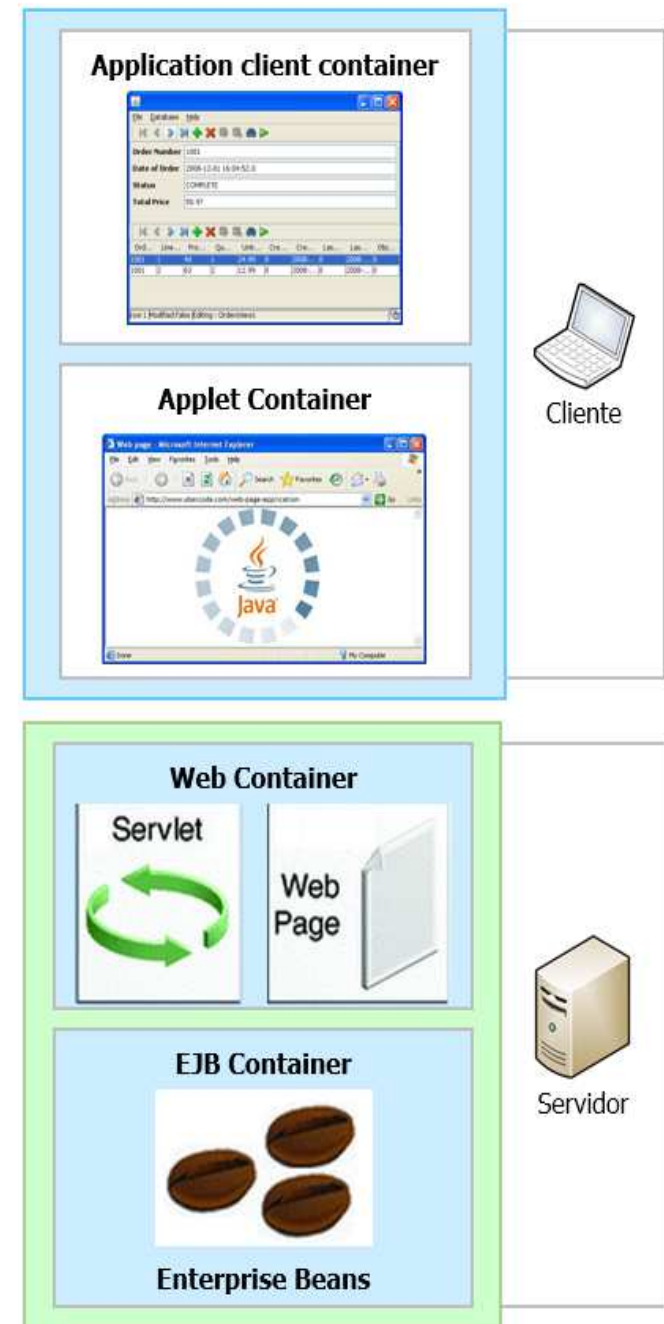
- roda no cliente
- carrega os componentes do servidor (JAR)
- composto de
 - ✓ browser + applet viewer



Tipos de Containers Java EE


■ Web

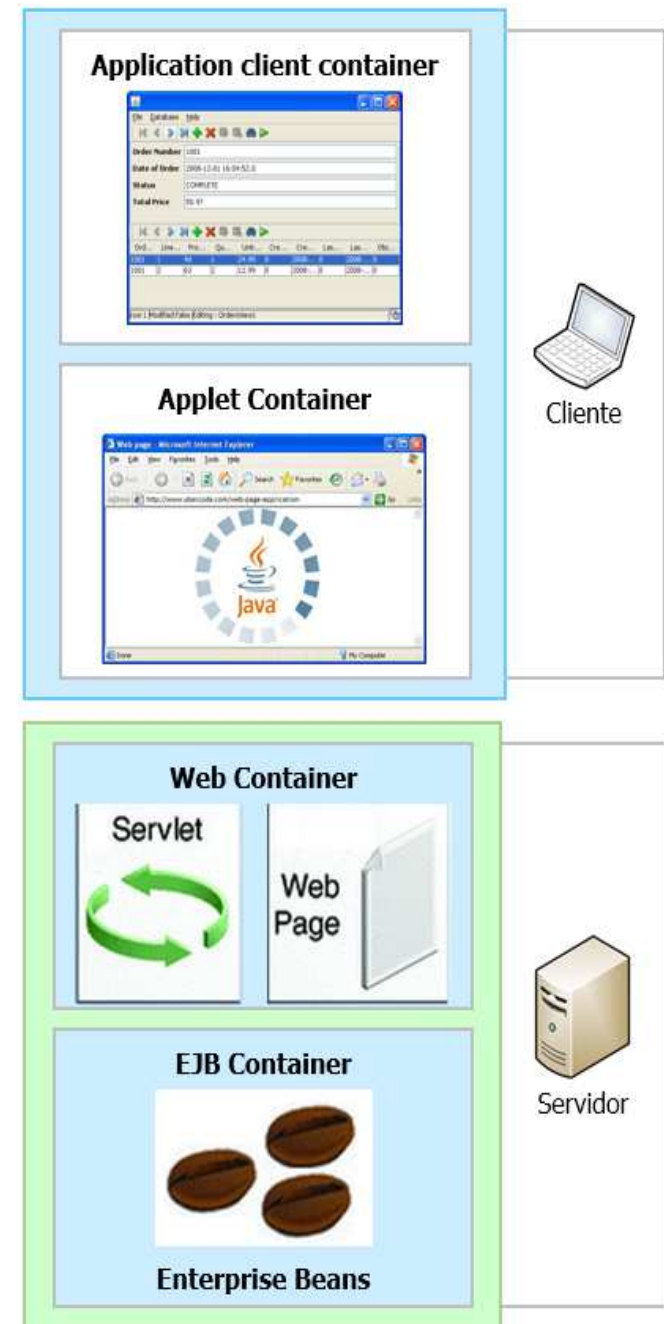
- roda no servidor (camada web)
- **controla o ciclo de vida de servlets**
- servlet container
- gerencia a execução de páginas web
 - JSP ou JSF
- pode gerenciar enterprise beans (**JEE6**)
 - ✓ EJB Lite
- intercepta requests **HTTP**, SMTP, FTP
- composto de
 - ✓ JavaSE (JRE)
 - ✓ JavaEE
 - segurança, concorrência, persistência, mensageria, webservices, injeção de dependência, validação e recursos
- implementações
 - Apache Tomcat e Jetty



Tipos de Containers Java EE

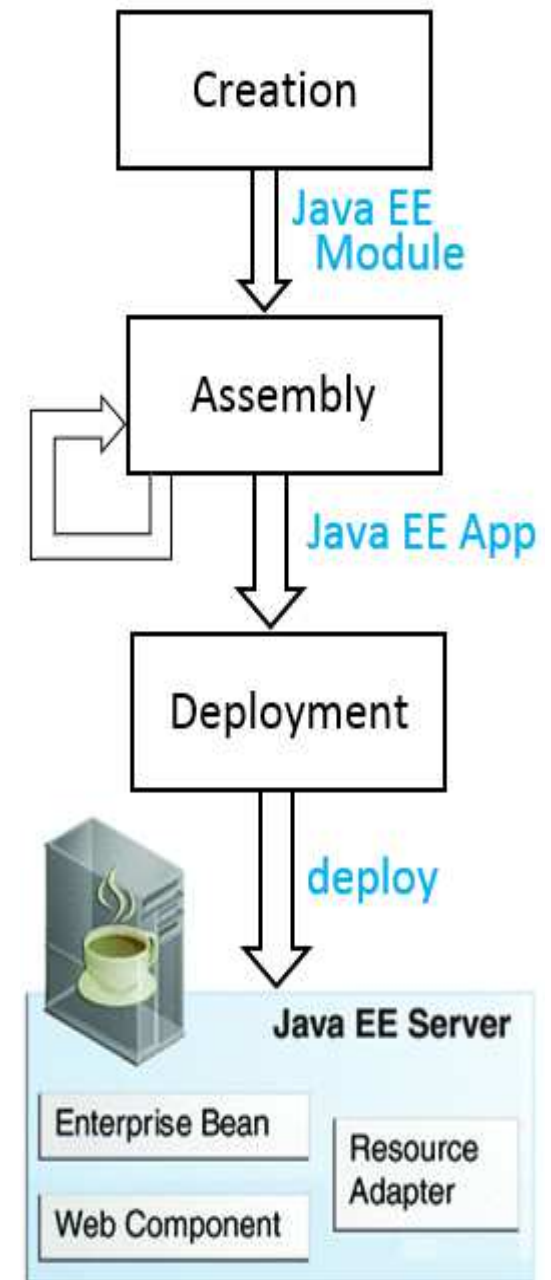
■ Enterprise JavaBeans

- roda no servidor
- camada de negócios
 - ✓ componentes executam a lógica de negócio
- gerencia o ciclo de vida dos EJBs
 - pooling (instanciar,destruir), caching
 - **sincronização de estado (persistência) J2EE**
- composto de
 - ✓ JavaSE (JRE)
 - ✓ JavaEE
 - segurança, transação, concorrência, persistência, mensageria, webservices, injeção de dependência, validação e recursos
- implementações
 - ✓ servidores de aplicação full (JBoss, GlassFish)
 - ✓  **OpenEJB**



Deployment de Aplicações JEE

- Instalar aplicação no servidor JEE
 - publicar componente no seu container
 - componentes são agrupados em módulos
- **Plataforma JEE (JSR 244, 316 e 342)**
 - define requisitos para montagem da aplicação
 - ciclo de vida da aplicação
 - deployment
 - ✓ subprocessos
 - ⇒ instalação, configuração e execução
 - ✓ 3 tipos de unidades de deployment
- **JSR 088**
 - requisitos para deployment (fornecedores)
 - ✓ servidor de aplicação (Java EE Product Provider)
 - ✓ IDE (Tool Provider)
 - 2 subprocessos de deployment
 - ✓ instalação e configuração



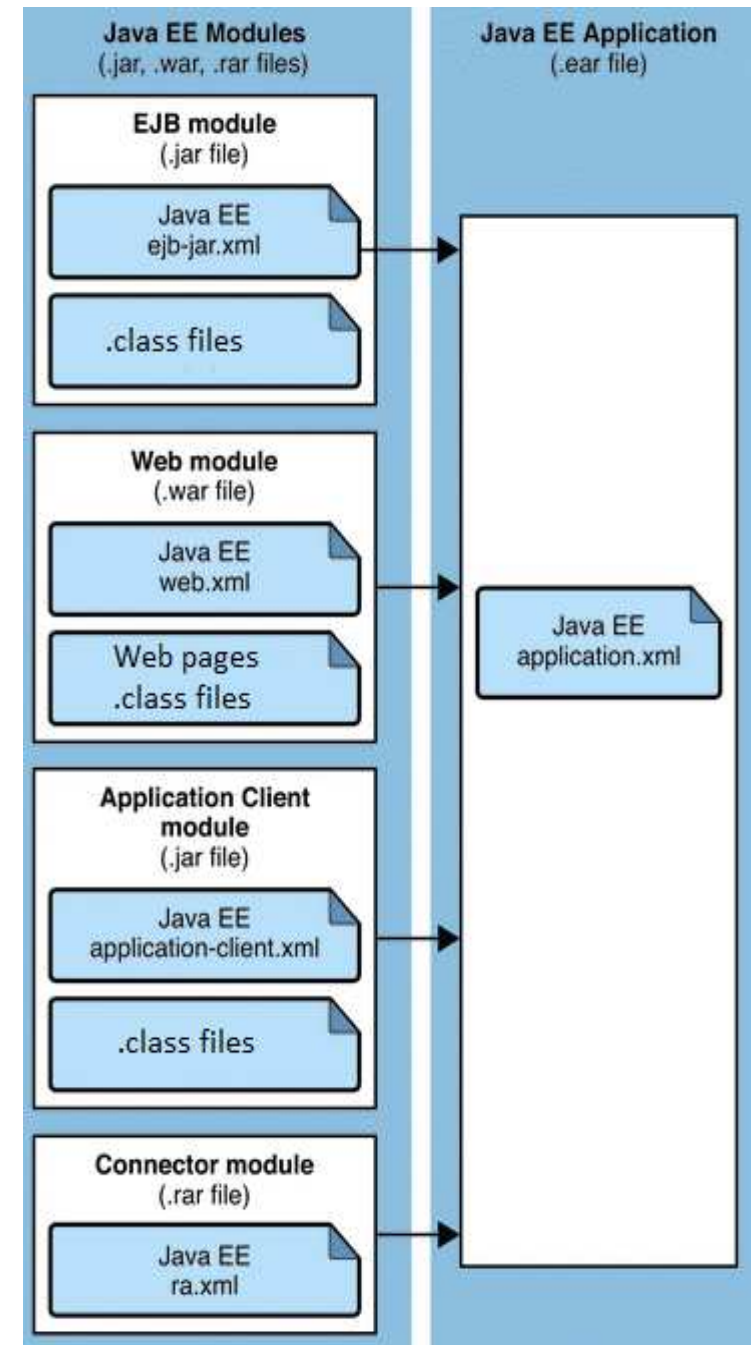
Estrutura de Aplicações JEE

▪ Conjunto de módulos

- módulo
 - ✓ unidade básica
 - ✓ agrega componentes ou outros módulos
 - ✓ empacotado em um Java ARchive (JAR)
 - ✓ arquivos de meta-informação (**opcional**)
→ descritores de implantação (DD)
- permite a portabilidade
 - ✓ módulos podem compor diferentes aplicações
 - ✓ dispensa codificação extra
- Tipos de módulos
 - ✓ EAR, WAR, JAR e RAR

▪ Distribuição de aplicações

- Enterprise ARchive (EAR)
- Web ARchive (WAR)
- Java ARchive (JAR)



Estrutura de Aplicações JEE

▪ Enterprise ARchive (EAR)

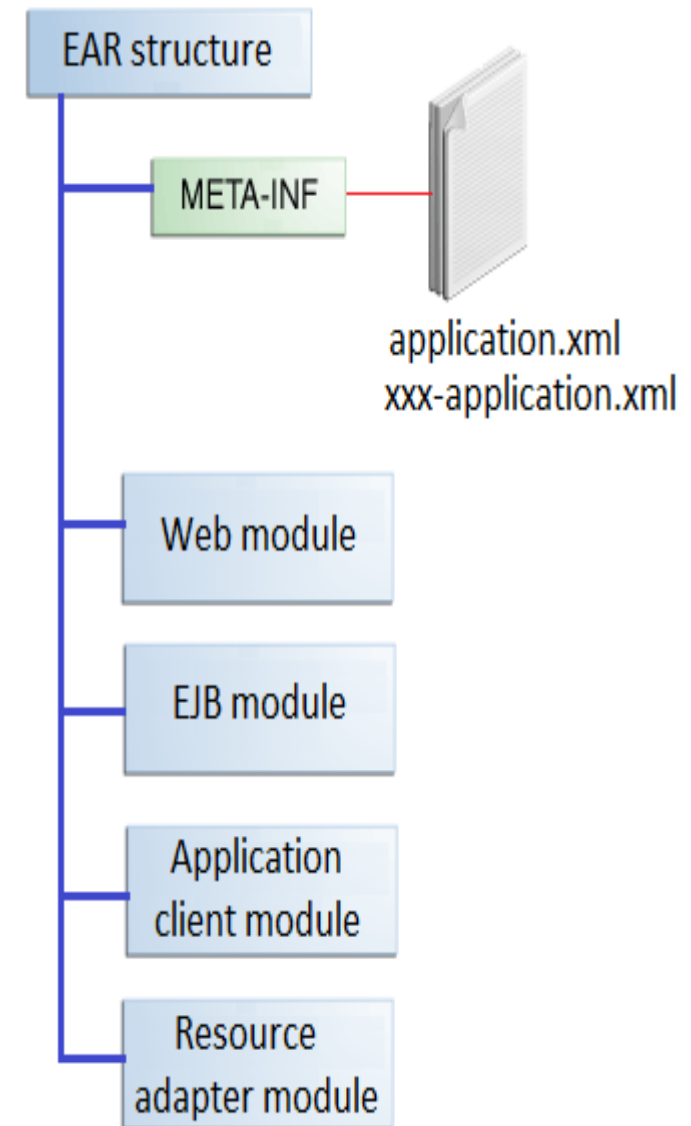
- ✓ é um arquivo JAR padrão com extensão .ear
- ✓ contém os módulos JEE da aplicação

• módulos JEE

- ✓ componentes para um tipo de container
- ✓ descritor de implantação (**opcional**)
 - XML com as configurações dos componentes
 - informações declarativas
 - alteração sem modificar código Java
 - lidos em tempo de execução pelo servidor
 - sobrescreve annotations (JEE7)
 - dois tipos:
 - descritor JEE e descritor de runtime

• tipos de módulos JEE

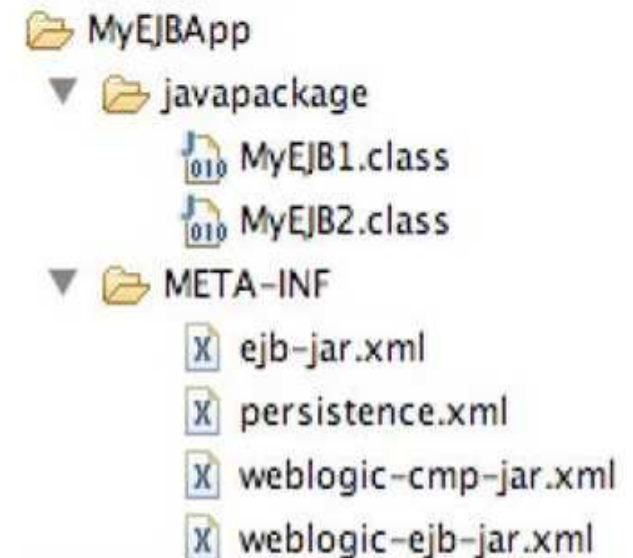
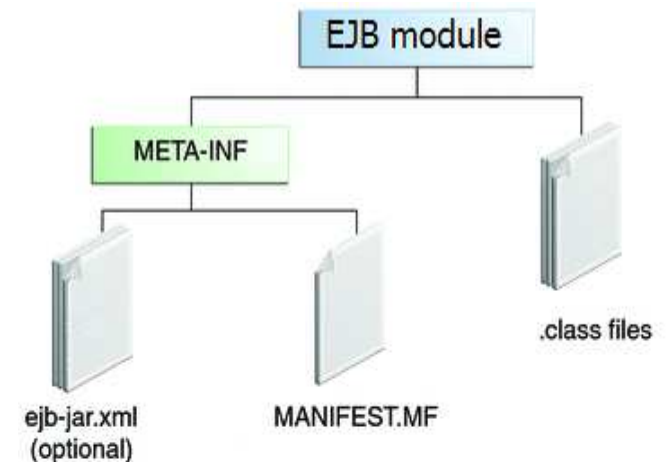
- ✓ módulos EJB
- ✓ módulos Web
- ✓ módulos de aplicações clientes
- ✓ módulos de adaptadores de recursos



Estrutura de Aplicações JEE

■ Módulo EJB

- arquivo JAR com extensão .jar
 - ✓ arquivo ZIP
- estrutura
 - ✓ META-INF (**opcional**)
 - descritores de implantação
 - ejb-jar.xml (JEE)
 - xxxx-ejb-jar.xml (runtime)
 - manifesto
 - metadados sobre o empacotamento
 - formato -> pares de atributo: valor
 - atributo Main-Class (JAR executável)
 - atributo Class-Path (**dependências**)
 - assinatura e java-beans
 - ✓ pacotes de classes (raiz)
 - Enterprise Beans (classes)
 - classes e interfaces



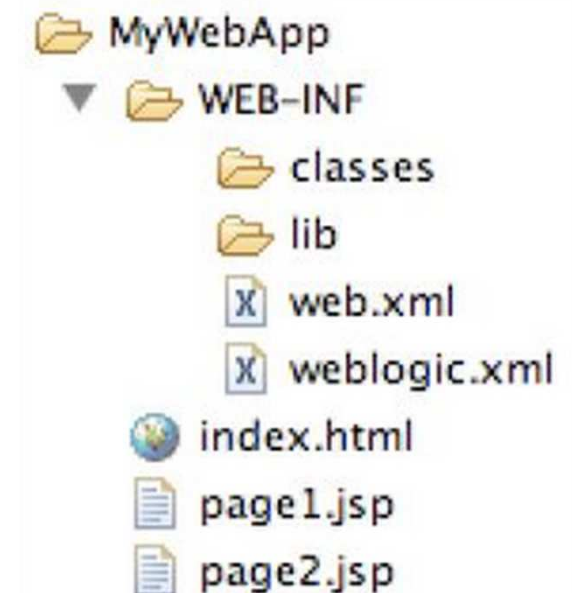
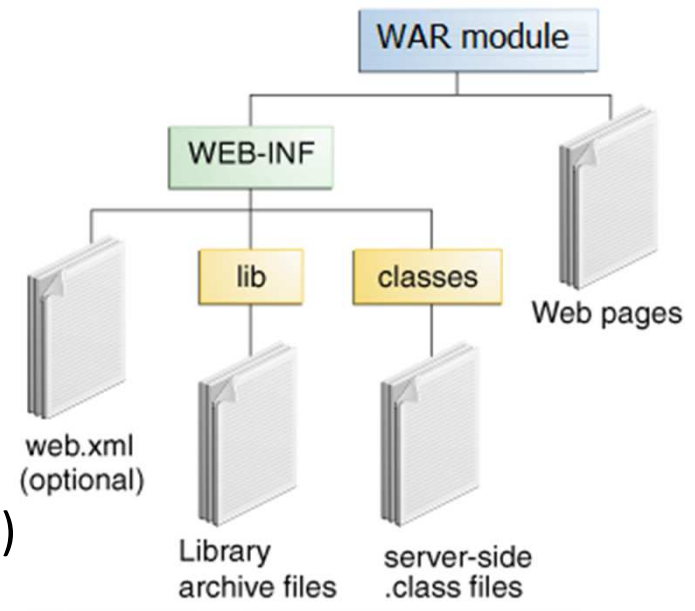
Estrutura de Aplicações JEE

■ Módulo Web

- ✓ arquivo JAR com extensão .war (Web ARchive)
- ✓ corresponde à aplicação web
- ✓ composto de
 - servlets, páginas web (JSP, JSF) (**dinâmico**)
 - recursos web (xhtml, imagens, etc) (**estático**)
 - classes utilitárias

• estrutura

- ✓ raiz
 - páginas web, classes clientes, imagens, etc
- ✓ WEB-INF
 - descritores de implantação
 - xxxx.xml (runtime)
 - web.xml (JEE) (**opcional?**)
 - obrigatório
 - JEE5: servlets, filters ou listeners
 - JEE6: JSF, segurança, sobrescrever annotations



Estrutura de Aplicações JEE

■ Módulo Web

- estrutura

- ✓ WEB-INF

- subdiretórios

- **classes**: servlets, EJB, beans, classes utilitárias

- **lib**: arquivos JAR (bibliotecas, módulo EJB)

- **tags**: bibliotecas de tags (**JEE5**)

- JEE6: fica em web-inf

- ✓ Enterprise Beans (JEE6)

- fornece a lógica de negócio da aplicação web

- pode ser empacotado em um WAR

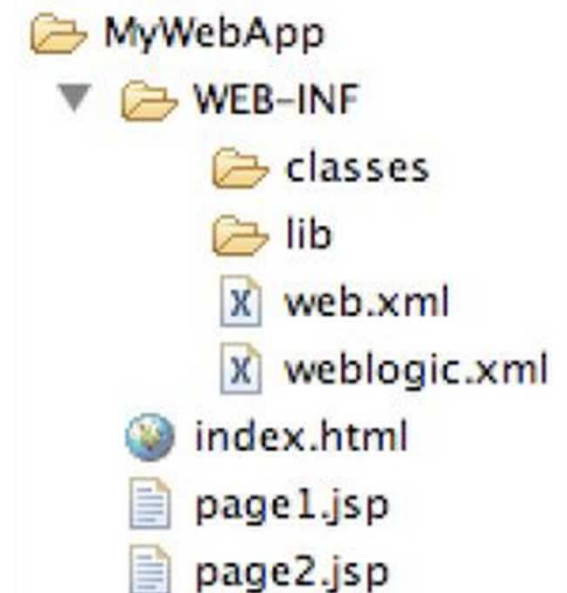
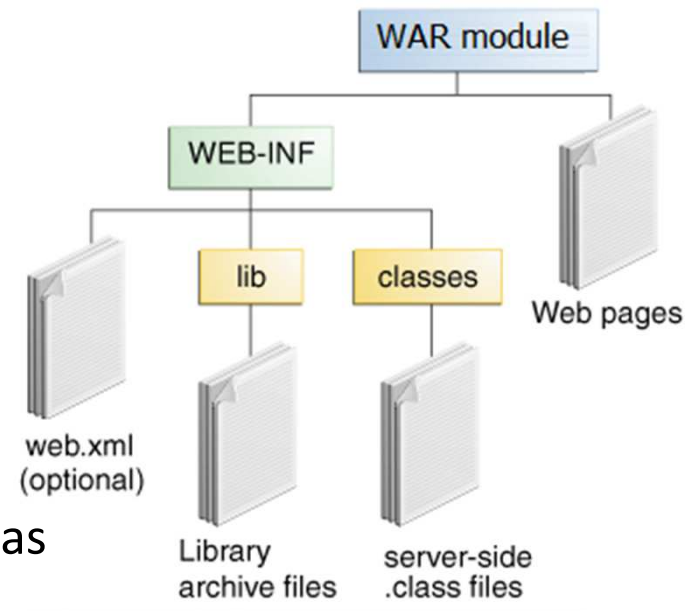
- simplifica deployment

- classes individuais ([web-inf/classes](#))

- pacote JAR ([web-inf/lib](#))

- não requer ejb-jar.xml em meta-inf (web-inf)

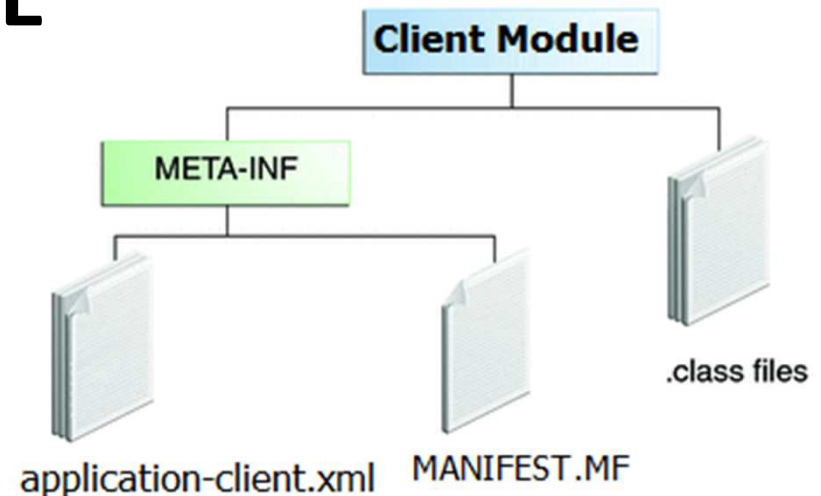
- não é considerado módulo EJB (EJB JAR)



Estrutura de Aplicações JEE

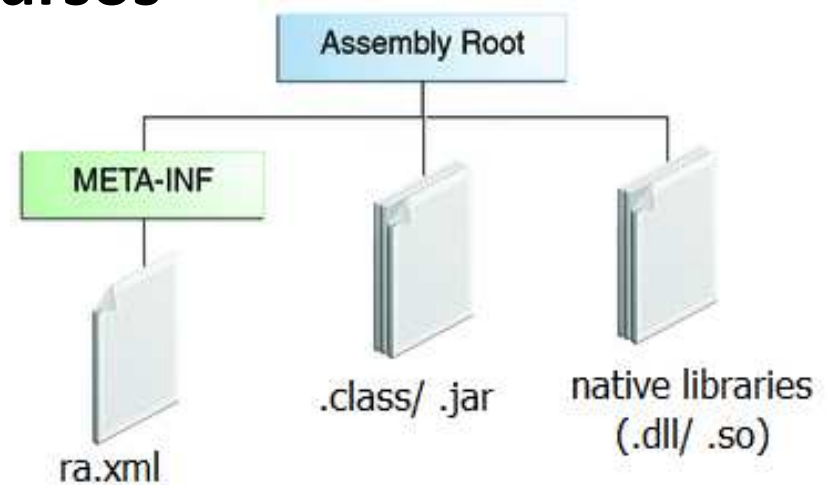
■ Módulo de Aplicação cliente

- ✓ arquivo JAR com extensão .jar
- ✓ standalone que roda no cliente
- ✓ estrutura
 - ✓ raiz: classes e bibliotecas clientes
 - ✓ META-INF
 - descritores de implantação
 - application-client.xml (JEE) e xxxx.xml (runtime)
 - manifesto: atributo Main-Class



■ Módulo de Adaptadores de recursos

- ✓ arquivo JAR com extensão .rar
- ✓ implementa um Connector para um EIS
- ✓ estrutura
 - ✓ raiz: classes, interfaces e bibliotecas
 - ✓ META-INF (opcional)
 - descritores de implantação
 - ra.xml (JEE) e xxx.xml (runtime)



[21] CESGRANRIO - 2013 - BNDES

O modelo de programação Java Enterprise Edition é baseado em containers, que fornecem todos os serviços necessários para a aplicação corporativa. Em consequência,

- a) uma página JSP é compilada pelo Container toda vez que é acessada por um usuário, assim como servlets.
- b) a camada cliente roda os componentes Web da arquitetura Java EE, como Java Server Pages e Servlets.
- c) o desenvolvedor pode escrever menos código, diminuindo o tempo de desenvolvimento, os riscos de projeto e os problemas de manutenção.
- d) as aplicações corporativas do modelo Java EE podem ser vistas como um modelo de camada única composto de classes especializadas da classe `javaServerFaces`.
- e) os Servlets passam a ser classes dependentes da classe `javaServerFaces`, podendo ser usadas em aplicações JEE5.

[22] FGV - 2013 - ALEMA

De acordo com a especificação da tecnologia Java EE 6 (*Enterprise Edition*), um container de serviços pode ser configurado

- a) por meio de métodos especiais de configuração dos *Beans Java* de acesso.
- b) usando descritores de implantação (*deployment descriptors*) e/ou por meio de anotações no código.
- c) por meio de arquivos de propriedade de recursos.
- d) somente por meio de anotações.
- e) por meio da configuração administrativa dos servidores de aplicações (*Enterprise Service Bus*).

[23] CESPE - 2012 - PEFOCE

Em uma arquitetura JEE distribuída, um contêiner representa um ambiente de execução padronizado que fornece serviços específicos a determinado componente. Um contêiner EJB, por exemplo, destina-se a prover a infraestrutura necessária para a execução de componentes que executem funcionalidades que realizam a lógica de negócio e dados específicos de determinada aplicação.

CESPE - 2005 - TCU

Um contêiner da Java 2 *Platform Enterprise Edition* (J2EE) é uma entidade de *software* executada no servidor e fornece o ambiente de execução para os componentes J2EE desenvolvidos.

[24] CONSULPLAN - 2012 - TSE

JavaEE v6 requer uma quantidade significativa de API's para uso pelas aplicações JavaEE, começando com o próprio núcleo das API's e incluindo muitas outras tecnologias Java. Componentes de aplicação JavaEE executam em ambientes fornecido por “*containers*”, que são parte da plataforma JavaEE. Essa plataforma suporta quatro tipos de “*containers*” correspondendo aos tipos de aplicações dos componentes. Além do “*application client containers*”, os outros três tipos são

- a) “*applet containers*”, “*web containers for servlets and JSP pages*” e “*enterprise bean containers*”.
- b) “*javascript containers*”, “*web containers for HTML and JSP pages*” e “*ultimate eclipse containers*”.
- c) “*applet containers*”, “*web containers for HTML and CSS pages*” e “*enterprise eclipse containers*”.
- d) “*javascript containers*”, “*web containers for servlets and CSS pages*” e “*ultimate bean containers*”.

[25] FAPEU - 2005 - TRE-SC

Qual das alternativas abaixo é **INCORRETA** em relação a uma aplicação J2EE?

- a) Há quatro tipos de componentes J2EE: aplicações cliente, applets, componentes web e componentes EJB.
- b) Aplicações J2EE dependem da existência de um contêiner EJB (EJB container).
- c) Contêineres web (web containers) e contêineres EJB (EJB containers) podem ser executados na mesma Máquina Virtual Java.
- d) Alguns dos serviços padrão utilizados por uma aplicação J2EE são fornecidos pela plataforma J2SE.

[26] CESGRANRIO - 2006 - DECEA

Sobre a arquitetura J2EE, assinale a afirmação correta.

- a) Servlets e arquivos JSP são executados no WEB Container.
- b) Applets e Servlets são compilados e executados no servidor.
- c) Applets devem acessar diretamente o banco de dados, obtendo melhor desempenho.
- d) A tecnologia interoperável para o uso de componentes distribuídos é dbCOM+.
- e) O EJB Container exige o sistema operacional Linux ou UNIX para execução.

[27] CESGRANRIO - 2013 - BNDES

Ao instalar uma aplicação Java EE, a ferramenta de instalação deve ler o descritor de instalação de aplicação do arquivo .ear da aplicação, que é encontrado em

- a) META-INF/application.xml
- b) META-INF/application-client.xml
- c) META-INF/deployment.xml
- d) META-INF/ejb-jar.xml
- e) META-INF/MANIFEST.ML

[28] FCC - 2012 - TJ-PE

Sobre a plataforma Java EE 6, é correto afirmar:

- a) Simplifica a implantação sem a necessidade de descritores de implantação, com exceção do descritor de implantação exigido pela especificação servlet, o arquivo web.xml.
- b) Necessita do descritor de implantação ejb-jar.xml e entradas relacionadas aos *web services* no arquivo web.xml.
- c) Faz uso de anotações (*annotations*). Anotações são modificadores Java, semelhantes aos públicos e privados, que devem ser especificados nos arquivos de configuração XML.
- d) A especificação EJB 3, que é um subconjunto da especificação Java EE, define anotações apenas para o tipo *bean*.
- e) Anotações são marcados com um caracter # (cerquilha).

[29] CESPE - 2008 - TCU

... *figura + texto* ...

Considere que, durante a liberação do primeiro *build* do CMS, tenham sido produzidos diversos módulos instaláveis, aderentes à plataforma J2EE, para *deploy* em servidores de aplicação. Nessa situação, é correto afirmar que, caso todos os módulos de *software* contenham a terminação WAR, isso sugere que a tecnologia EJB não foi utilizada para implementação do CMS.

[30] CESGRANRIO - 2008 - BNDES

Uma aplicação empresarial contendo componentes EJB e módulos web deverá ser publicada em um servidor de aplicações compatível com J2EE. No contexto do empacotamento dessa aplicação para publicação (deploy), é correto afirmar que

- a) não há como juntar componentes EJB e módulos web em uma mesma aplicação, pois deverão ser publicados separadamente.
- b) um arquivo EAR poderá conter arquivos WAR e JAR representativos dos módulos web e EJB.
- c) o tamanho do pacote, em bytes, sempre fica maior que o código original, em virtude do algoritmo empregado no empacotamento da aplicação em um arquivo EAR.
- d) módulos web não devem ser empacotados, pois isso inviabiliza seu acesso pela Internet.
- e) arquivos JAR servem apenas para empacotar componentes EJB.

Gabarito

[01] - E E C	[11] - A	[21] - C
[02] - C	[12] - E	[22] - B
[03] - E C E	[13] - E	[23] - C Anulada
[04] - C	[14] - B	[24] - A
[05] - C E E	[15] - A	[25] - B
[06] - C C	[16] - E	[26] - A
[07] - B	[17] - C	[27] - A
[08] - E E	[18] - E E C	[28] - A
[09] - E	[19] - E	[29] - C
[10] - E	[20] - E	[30] - B



Java Enterprise Edition



Leonardo Marcelino

<http://www.itnerante.com.br/profile/LeonardoMarcelino>