

JAVA EE - CONTAINERS JAVA EE - QUESTÕES TEORI

Estamos abordando uma visão geral sobre a arquitetura do JAVA EE, como ela é formada, camadas, modelos, aplicações, componentes e a parte dos containers (espécie de recipiente para que você possa colocar os componentes específicos).

Na camada cliente temos dois containers: aplicação cliente e applet. Vamos abordar mais detalhadamente durante o nosso curso.

Tópicos que iremos estudar em nossas aulas:

- Introdução
- Conceituação
- Exemplos de serviços oferecidos
- Tipos
- Rodando no servidor Java EE e no cliente
- Camadas
- Componentes Java EE
- Camadas, containers e componentes Java EE
- Esquema 1
- Esquema 2
- Configuração de containers
- Montagem de componentes nos containers
- Questões de concursos
- Referências

CONCEITUAÇÃO

Os containers são o conhecimento central da arquitetura Java EE, os quais são formados por componentes que rodam nos containers.

Os três C's das aplicações: componentes, containers e conectores.

São uma interface entre o componente e funcionalidade de baixo nível específico da plataforma que suporta o componente.



5m

ANOTAÇÕES

| |
|--|
| |
| |
| |
| |

Viu algum erro neste material? Contate-nos em: degravacoes@grancursosonline.com.br

Dessa forma, temos um servidor ou um cliente. No cliente há dois containers (aplicação cliente e applet) e no servidor também (container web e EJB). Por conseguinte, temos os componentes web rodando no container web e container EJB, e as aplicações clientes rodando no container aplicação cliente, e os applets no container applet.

Tudo isso é oferecido pelos containers que abrigam componentes Java EE e liberam os componentes da interação com a infraestrutura subjacente.

Sua funcionalidade é definida pela plataforma JAVA EE e é diferente para cada tipo de componente, porque temos tipos diferentes de containers.

No entanto, o servidor permite que os diferentes tipos de componentes trabalhem juntos para fornecer funcionalidade de uma aplicação corporativa.

Isso porque uma aplicação corporativa é formada por componentes, sendo componentes clientes, web, negócio, então o servidor permite que diferentes tipos de componentes trabalhem juntos, embora cada um esteja em um container específico.

Oferecem serviços para os componentes de forma transparente para o desenvolvedor.

Exemplos de serviços oferecidos pelos containers para os componentes:

- Segurança
- Ciclo de vida
- Gerenciamento de transação
- Persistência
- Controle de concorrência

TIPOS

- Container de Aplicação Cliente que roda no cliente.
- Container de Applet que roda no cliente.
- Container Web que roda os componentes Web.
- Container EJB que roda os componentes de negócio EJB.

O Container de Aplicação cliente gerencia a execução de componentes de aplicações clientes de forma bem transparente.

O Container de Applet gerencia a execução de applets e consiste em duas partes: em execução juntos no cliente, em um navegador da Web e Java Plug-in para rodar os applets.

ANOTAÇÕES

| |
|--|
| |
| |
| |
| |



Já na parte do servidor, o Container Web gerencia a execução de páginas JSP, Servlets, e alguns componentes EJB.

Além disso, o Container EJB gerencia a execução de EJB's.

RODANDO NO SERVIDOR JAVA EE E NO CLIENTE

Rodam no cliente:

- Container de Aplicação cliente
- Containter de Applet

Rodam no servidor Java EE:

- Container Web
- Container EJB

CAMADAS

- Máquina do cliente

Camada Cliente

- Servidor Java EE

Camada Web

Camada de Negócio

É uma camada intermediária.

- Máquinas onde residem bancos de dados e sistemas legados

Camada EIS

Enterprise information system

Sistemas de informação corporativa

ANOTAÇÕES

| |
|--|
| |
| |
| |
| |

COMPONENTES JAVA EE

- Componentes no cliente

Aplicações clientes

Applets

- Componentes web no servidor

Servlet

JSP

JSF

- Componentes no servidor

EJB

CAMADAS, CONTAINERS E COMPONENTES JAVA EE

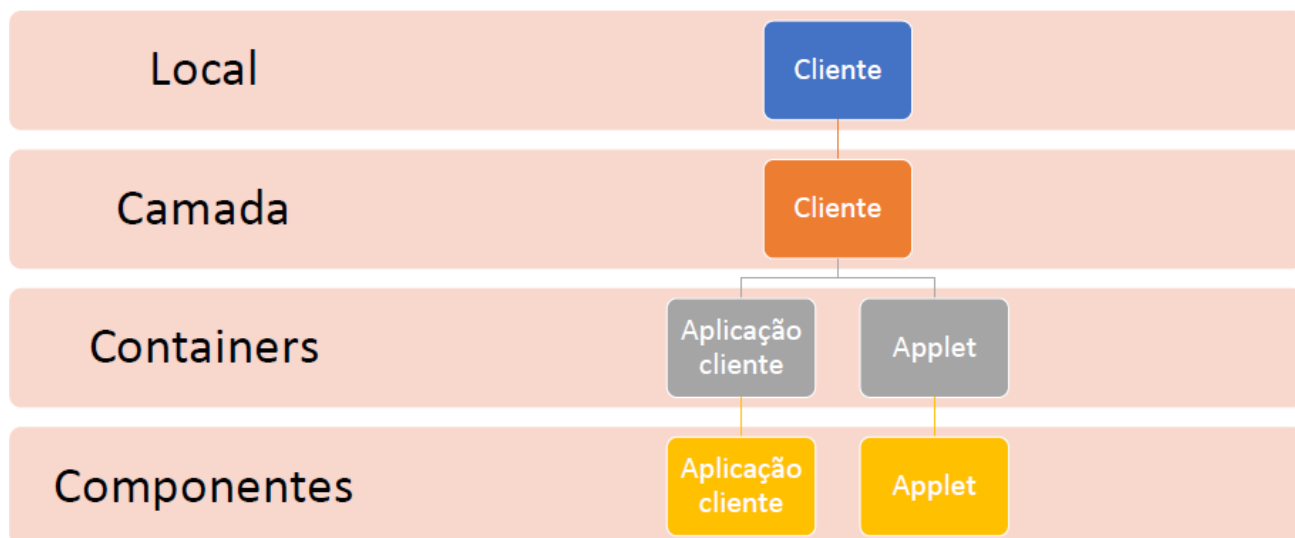
| Local | Camada | Container | Componente |
|----------|---------|-------------------|-------------------|
| Cliente | Cliente | Aplicação cliente | Aplicação cliente |
| | | Applet | Applet |
| Servidor | Web | Web | Servlet |
| | | | JSP |
| | | | JSF |
| | Negócio | EJB | EJB |

Em resumo: temos que:

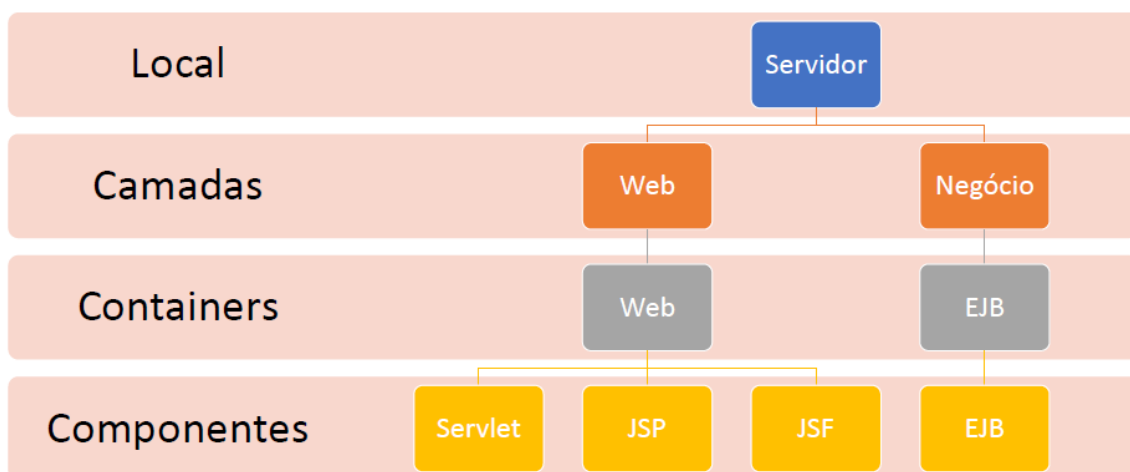
ANOTAÇÕES

| |
|--|
| |
| |
| |
| |

Viu algum erro neste material? Contate-nos em: degravacoes@grancursosonline.com.br



Por outro lado, o servidor:



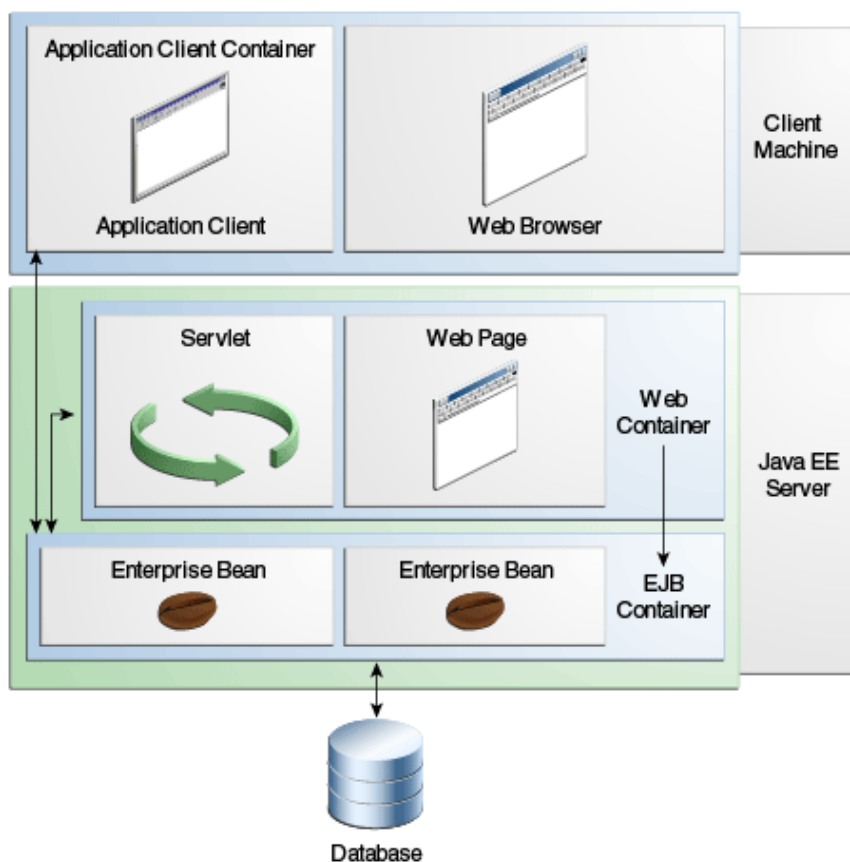
ANOTAÇÕES

| |
|--|
| |
| |
| |
| |

Viu algum erro neste material? Contate-nos em: degravacoes@grancursosonline.com.br

15m

Esquema 1:



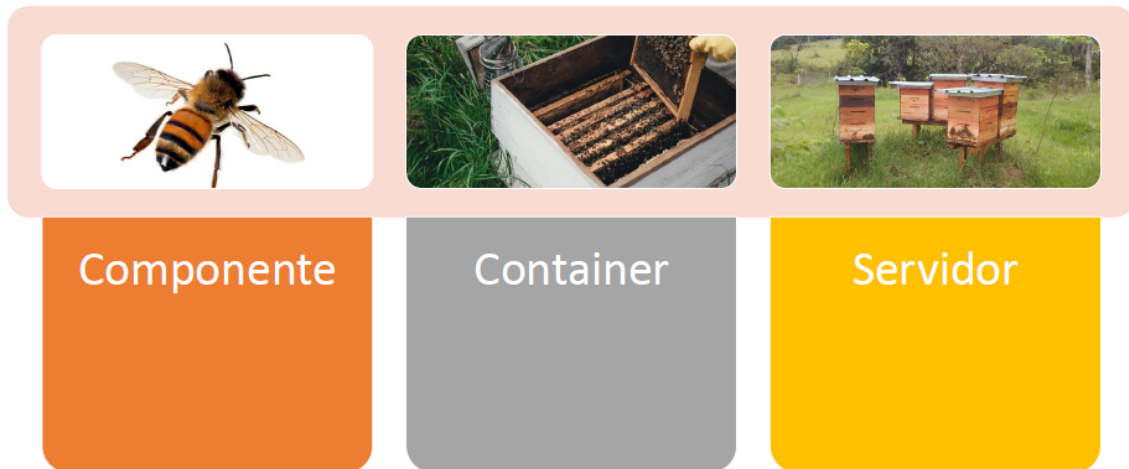
Podemos ter a seguinte visão:

ANOTAÇÕES

| |
|--|
| |
| |
| |
| |

Viu algum erro neste material? Contate-nos em: degravacoes@grancursosonline.com.br

Esquema 2:



Poderá servidor ou cliente, ou servidor cliente, pois no cliente nós temos dois containers. Assim: componente abelha roda de uma colmeia específica; o conjunto de colmeias vai ser o servidor ou cliente. Cada um desses dois rodam dois tipos de containers.

CONFIGURAÇÃO DE CONTAINERS

Personaliza o suporte fundamental fornecido pelo servidor Java EE, incluindo serviços como:

- Segurança
- Gerenciamento de transações
- Pesquisas com JNDI

Java Naming and Directory Interface

- Conectividade remota

Pode ser feita em tempo de implantação com programação declarativa. É uma mudança de atributos.

Exemplos de itens que são configurados do container em tempo de implantação:

ANOTAÇÕES

| |
|--|
| |
| |
| |
| |

- Estabelecer segurança, tratamento transacional.
- Mapear nomes entre: a aplicação, os recursos disponíveis.

Na parte de Segurança, temos o modelo de segurança Java EE que permite configurar um componente da web ou *enterprise bean*, para que os recursos do sistema sejam acessados apenas por usuários autorizados.

Há, ainda, o gerenciamento de transações do modelo de Transação Java EE que permite especificar relacionamentos entre métodos que compõem uma única transação, para que todos os métodos em uma transação sejam tratados como uma única unidade.

Temos as seguintes partes:

- Pesquisas com JNDI;
- Serviços de Pesquisa JNDI que fornecem uma interface unificada para vários serviços de nomes e diretórios, para que os componentes da aplicação possam acessar esses serviços.

Além disso, a Conectividade remota, Modelo de Conectividade Remota Java EE gerencia as comunicações de baixo nível entre clientes e *beans* corporativos. Depois que um *bean* corporativo é criado, um cliente chama métodos nele como se estivesse na mesma máquina virtual.

MONTAGEM DE COMPONENTES NOS CONTAINERS

Antes de um componente Java EE ser executado ele deve ser:

Montado em um módulo Java EE e Implantado em seu container.

O Processo de montagem (Assembly process) envolve a especificação de configurações de container para: cada componente na aplicação Java EE e a própria aplicação Java EE em si.

A arquitetura Java EE fornece serviços configuráveis. Os componentes dentro de uma mesma aplicação Java EE podem se comportar de forma diferente, dependendo de onde eles estão implantados:

Por exemplo, um EJB pode ter definições de segurança que o permite ter:

- Um certo nível de acesso a um banco de dados no ambiente de produção.



20m

ANOTAÇÕES

| |
|--|
| |
| |
| |
| |

- Outro nível de acesso a um banco de dados em outro ambiente de produção.

O container também gerencia os serviços não configuráveis, tais como: ciclo de vida de EJB's, Servlets, Pooling de recursos de conexão de banco de dados, persistência de dados, acesso às API's da plataforma Java EE.

Na próxima aula analisaremos as questões de concursos pertinentes ao tema da nossa aula.

Este material foi elaborado pela equipe pedagógica do Gran Cursos Online, de acordo com a aula preparada e ministrada pelo professor Rogério Gildo Araújo.

A presente gravação tem como objetivo auxiliar no acompanhamento e na revisão do conteúdo ministrado na videoaula. Não recomendamos a substituição do estudo em vídeo pela leitura exclusiva deste material.

ANOTAÇÕES

| |
|--|
| |
| |
| |
| |