

JAVA EE - MODELO DE APLICAÇÕES JAVA EE - TEORIA

Tópicos

- Introdução;
- Modelo de aplicações Java EE;
- Camadas;
- Esquema;
- Camada Cliente;
- Camada Web;
- Camada de Negócios;
- Camada EIS;
- Aplicações corporativas;
- Segurança;
- Questões de concursos;
- Referências.

Introdução

A plataforma Java EE utiliza um modelo de aplicação **distribuída e multicamadas** para **aplicações corporativas**.

A lógica das aplicações é **dividida em componentes** de acordo com a **função**.

Os componentes podem ser **instalados em várias máquinas dependendo do nível no ambiente de várias camadas** Java EE ao qual o componente de aplicação pertence.



5m

- Há a camada cliente, sendo que é possível ter alguns componentes clientes que acesam os dados de um servidor Java EE. Em um servidor Java EE, é possível ter duas camadas: a **camada web e a camada de negócio**. Então, o servidor Java EE vai representar uma **camada intermediária**, que vai abrigar duas camadas. Ainda há a **camada Enterprise Information System, ou sistema de informações corporativas**, que pode ser de dados e outros tipos de informações, uma vez que trata-se de um **repositor de informações para jogar os dados e para consumir os dados**. Nessas camadas, então, residirão os chamamos contêineres, de modo que, para a camada web, por exemplo, há o contêiner web, que vai rodar os componentes web, isto é, **para cada camada, há um contêiner específico que saiba trabalhar com os componentes**. Assim sendo, as **aplicações corporativas são formadas por componentes, e cada componente possui a sua função dentro da aplicação**.

Modelo de Aplicações Java EE

Começa com **linguagem Java** e **máquina virtual Java**.

- Então, é preciso ter a base bem sedimentada, ou seja, **trabalhando bem com Java SE**, para que consiga criar as aplicações.

A sua **base** é formada pelas características de:

- Portabilidade;
- Segurança;
- Produtividade.

Define uma arquitetura para implementação de serviços que proporcionam para **aplicações multicamadas**:

- Escalabilidade;
- Acessibilidade;
- Gerenciamento.

Particiona o **trabalho necessário** para implementar um serviço de **múltiplas camadas** nas seguintes partes:

- Os **serviços padrões** fornecidos pela **plataforma Java EE**;
 - Parte ou serviços de infraestrutura que o desenvolvedor irá utilizar.
- A lógica de **negócio** e de **apresentação** a ser implementada pelo **desenvolvedor**.

Diversas **complexidades inerentes** às aplicações corporativas são **construídas na plataforma** e **providas automaticamente pelos componentes** que o modelo suporta, alguns exemplos de complexidades:

- Gerenciamento de transações;
- Gerenciamento de ciclo de vida;
- Agrupamento de recursos.



10m

ANOTAÇÕES

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

O **desenvolvedor** poderá concentrar-se mais nos **detalhes da lógica de negócio** e **confiar na plataforma** para **fornecer soluções** para os problemas difíceis no nível do sistema de desenvolvimento de um serviço de várias camadas.

Camadas

São **áreas funcionais isoladas** que separam as funcionalidades de aplicações.

Aplicações Java EE são geralmente consideradas como **aplicações de três camadas**, porque, embora sejam divididas em três ou quatro camadas, são **distribuídas em três locais**:

- Na **máquina do cliente**;
- No **servidor Java EE**;
- Nas **máquinas** onde residem.
 - Bancos de dados;
 - Sistemas legados.

Então, é possível ter várias máquinas onde residem camadas diferentes da aplicação: máquina do cliente.

- Camada Cliente

Servidor Java EE

- Camada Web;
- Camada de Negócio.

Máquinas onde residem bancos de dados e sistemas legados:

- Camada EIS;
- *Enterprise Information System*;
- Sistemas de informação corporativa.



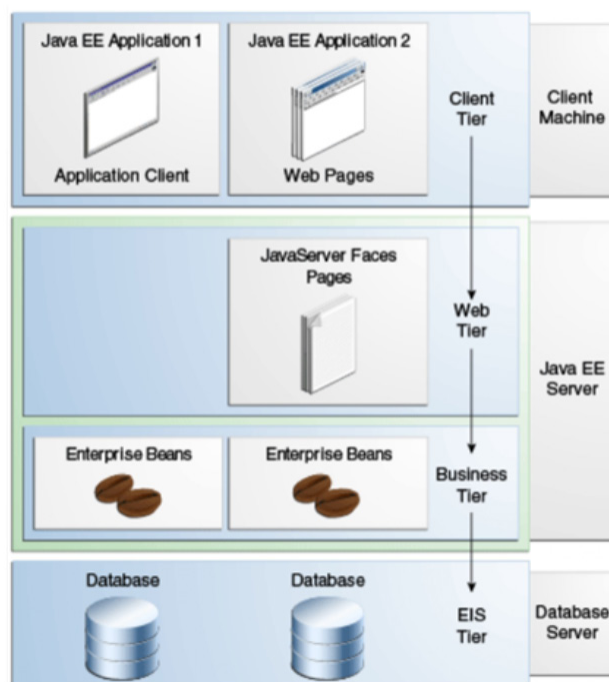
15m

ANOTAÇÕES

Camadas



Esquema



ANOTAÇÕES



Camada Cliente

Consiste em aplicações clientes que **acessam um servidor Java EE**. Geralmente, estão localizadas em uma **máquina diferente do servidor** e podem ser:

- Clientes Java EE;
- **Aplicações não Java;**
- Navegador web;
- Aplicações independentes;
- Outros servidores.

Os **clientes fazem requisições** ao servidor, de modo que o **servidor processa** as requisições e **retorna uma resposta ao cliente**.

- A camada web que processa as requisições e faz o retorno.

Camada Intermediária

É onde são realizadas as **funções de negócios** para dar **suporte** a vários usuários, **controlar e gerenciar melhor as aplicações**.

Representa um ambiente que é controlado de perto pela **área de TI** de uma empresa.

Geralmente, é executada em **hardware de servidor dedicado** e tem **acesso aos serviços completos da empresa**.

É composta pelas camadas:

- Web;
- De Negócios.

Lida com **requisições** do cliente.

- Camada web.

Processa dados da aplicação.

- Camada de negócios.



ANOTAÇÕES

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

Armazena dados, em um **banco de dados**, na camada de dados.

- **Camada de Negócios**

O desenvolvimento de aplicações Java EE **concentra-se nesta camada** para tornar o **gerenciamento de aplicações corporativas** mais:

- Fácil;
- Robusto;
- Seguro.

Camada Web

Consiste em componentes que lidam com a **interação** entre:

- Clientes;
- Camada de negócios.

Algumas **funções** são:

- Gerar dinamicamente conteúdo em vários formatos para o cliente;
- Coletar entrada de usuários da interface do cliente;
- Retornar resultados apropriados dos componentes na camada de negócios;
- Controlar fluxo de telas e as páginas no cliente;
- Manter o estado dos dados para a sessão de um usuário;
- Executar alguma lógica básica;
- Manter alguns dados temporariamente em beans gerenciados.

Alguns exemplos de **componentes de tecnologia**:

- JSF;
- Expression Language;
- Servlets;
- Injeção de dependência e contextos para Java EE.



30m

ANOTAÇÕES

JSF

JavaServer Faces.

- Em MVC.

É uma estrutura de **componente de interface com o usuário** para aplicações web que permite:

- Incluir componentes de **interface do usuário** (como campos e botões) em uma **página XHTML**;
 - Página Facelets.
- **Converter e validar dados** do componente da interface do usuário;
- **Salvar dados** do componente da interface do usuário em **repositórios de dados do servidor**.

Manter o estado do componente.

Expression Language

É um **conjunto de tags padrão** usadas nas **páginas Facelets** para **se referir** aos componentes Java EE.

Servlets

São classes Java que:

- Processam **requisições dinamicamente**;
- Constroem **respostas** geralmente para **páginas HTML**.

Injeção de Dependência e Contextos para Java EE

São um **conjunto de serviços contextuais** que **facilitam o uso de EJB's** pelos **desenvolvedores junto** com a **tecnologia JavaServer Faces em aplicações web**;

É um **padrão** de desenvolvimento de programas utilizado quando é necessário **manter baixo o nível de acoplamento** entre diferentes módulos de um sistema.

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

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.
