



UNIVERSIDADE FEDERAL DO CEARÁ - Campus Quixadá

Cursos: SI, ES, RC, CC e EC

Código: QXD0043

Disciplina: Sistemas Distribuídos

Enterprise Java Beans (EJB)

Prof. Rafael Braga

Introdução

- O que são Java Beans?
 - Definição de JavaBean: “a reusable software componente that can be manipulated in a builder tool”
 - Construtor Vazio
 - Propriedades privadas, acessadas por métodos get e set (getXyz, setXyz)
 - Não é necessário herdar de nenhuma classe em especial (POJO)
 - Deve ser serializável (permitir que seu estado seja salvo e carregado posteriormente)
-

Introdução

- O que são Enterprise Java Beans?
 - É um dos principais componentes da plataforma JEE.
 - É um componente do tipo servidor que executa no contêiner do servidor de aplicação.
 - O seu principal objetivo é favorecer um rápido e simplificado desenvolvimento de aplicações Java baseado em componentes distribuídos, transacionais, seguros e portáteis.
 - Componente que encapsula a lógica de negócio da aplicação e possibilita a construção de aplicações tratando de conceitos como: persistência, processamento de transações, eventos, segurança, controle de concorrência e chamada a métodos remotos.
-

Introdução

- Histórico
 - A primeira versão do EJB, a versão EJB 1.0, foi lançada em março de 1998 na conferência de desenvolvedores Java chamada JavaOne 1998.
 - Ao longo das versões sucessoras, EJB 1.1, 2.0 e 2.1, diversas alterações foram feitas e funcionalidades foram incluídas, tais como:
 - Compatibilidade com protocolos CORBA (RMI-IIOP).
 - Suporte à Web Services.
 - A criação de uma aplicação EJB parecia ser muito difícil devido à complexidade que a API apresentava para os desenvolvedores iniciantes.
-

Introdução

- Histórico
 - Tornou-se claro para os programadores que a API não era inteiramente utilizada pela maioria das aplicações.
 - A versão 3.0, além de incluir outras funcionalidades, buscou:
 - Solucionar este problema da complexidade.
 - Introduzir formas de lidar com persistência e transações de forma simplificada com Java Annotations.
 - Utilizar características de frameworks mais compactos como Hibernate e Spring.
 - A versão 3.0 é vista como praticamente nova.
 - Foi lançada em maio de 2005
-

Introdução

- Java Beans é um modelo elegante para construção de aplicações baseadas em componentes.
- Em geral, é usado para construção de aplicações locais/centralizadas.
- Comunicação orientada a eventos.
- Interface gráfica auxilia na criação de aplicações com ajuda do mouse.

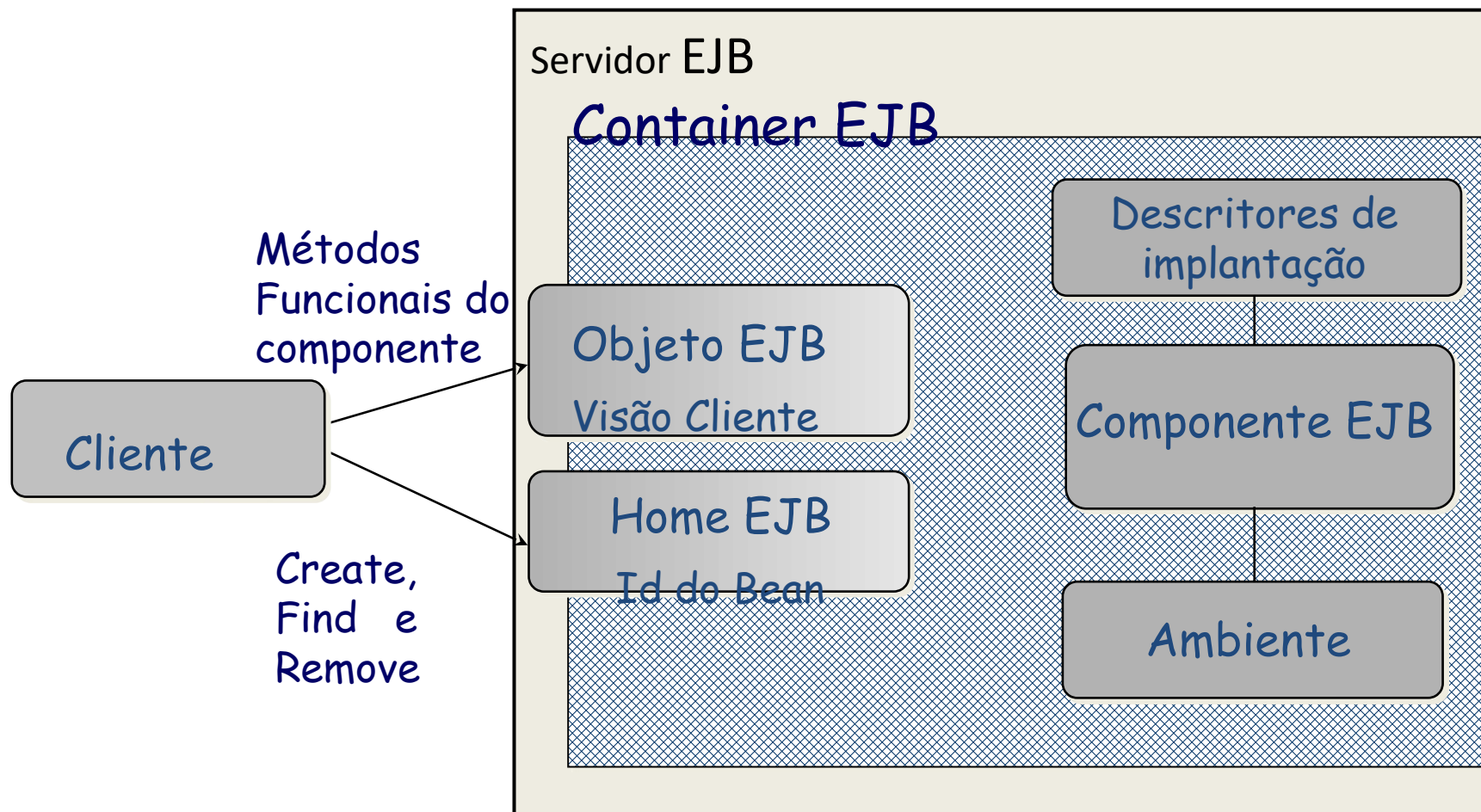
Objetivos de EJB

- Facilitar o desenvolvimento e gerenciamento de aplicações distribuídas
 - Distribuição
 - Persistência
 - Transações
 - **Principal objetivo** - aumentar a produtividade
 - Representa a lógica do negócio
 - Define interfaces remotas
 - Implementa a interface *EntityBean* ou *Session Bean*
 - Arquivo de propriedades
-

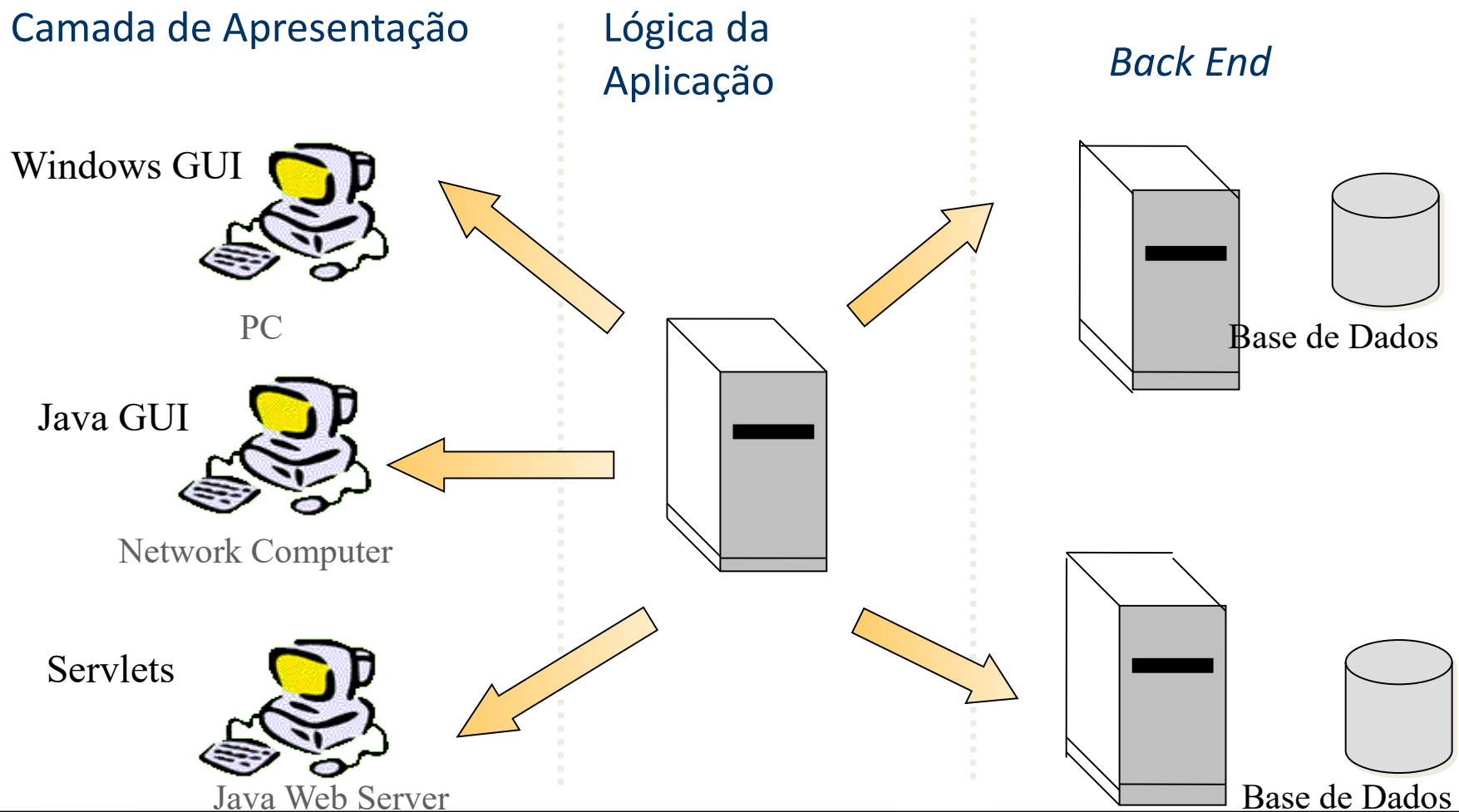
Enterprise Java Beans (EJB)

- ◆ Arquitetura de componentes para o desenvolvimento e implantação de *servidores* em ambientes distribuídos
- ◆ Tecnologia de Monitores de Transações de Componentes – CTM's, sustenta a arquitetura EJB
 - Infraestrutura para: gerenciamento automático de transações, distribuição de objetos, concorrência, segurança, gerenciamento de recursos, ampla população de usuários, trabalhos de missão crítica
- Outros serviços
 - Gerenciamento de estado
 - Ciclo de vida
 - Persistência

Arquitetura EJB



Arquiteturas de 3 camadas



Elementos da Arquitetura EJB

◆ Servidor EJB

- servidor de aplicação genérico que fornece um ambiente compatível com a especificação da arquitetura EJB
- fornece um ou mais containers para os componentes nele implantados
- responsável pelo gerenciamento e coordenação da alocação de recursos:
 - Threads, processos, memória, conexões a BD
 - serviços: transações, nomes, segurança e persistência

◆ Container EJB

- fornece contexto de execução e contexto transacional aos componentes
- registra o componente no serviço de nomes, cria e destrói instâncias
- fornece interface remota para o componente
- gerencia transações, estado e persistência

Elementos da Arquitetura EJB (*cont*)

- ◆ Componente EJB
 - Interface Home
 - ∅ define os métodos de ciclo de vida do componente: criação, remoção e busca
 - ∅ através dessa interface, clientes veem componentes EJB como uma coleção homogênea de instâncias
 - Interface Remote
 - ∅ define os métodos funcionais do componente
 - ∅ representa a visão que o cliente terá do componente
 - ∅ expõe todas as interfaces relacionadas à aplicação
 - Classe do Componente
 - ∅ implementa os métodos funcionais (aspectos funcionais do componente)
- ◆ Tipos de Componentes EJB:
 - Componentes Entidades (*Entity Beans*)
 - Componentes Sessão (*Session Beans*)

Papéis EJB

- ◆ Fornecedor de Componentes EJB
 - definição das interfaces home e remote
 - gera arquivo ejb-jar contendo 1 ou mais componentes
 - descritor de implantação
- ◆ Montador da Aplicação
 - gera arquivo ejb-jar com os componentes e suas instruções de montagem
- ◆ Implantador
 - geração das classes e interfaces adicionais que habilitam o container a gerenciar componentes EJB
- ◆ Fornecedor de Servidor EJB / Container EJB
 - ferramentas de implantação
 - suporte de tempo de execução para as instâncias dos componentes
 - API padrão para acesso aos serviços de transação, acesso a BD, ...
- ◆ Administrador do Sistema

Entity Beans

- ◆ Fornece uma visão em forma de objeto dos dados na BD (entidades)
- ◆ Permite acesso compartilhado por múltiplos usuários
- ◆ Possui “vida longa” → igual a do dado associado na BD
- ◆ Uma instância de um bean entidade, sua chave primária e sua interface remota sobrevivem a eventual quedas do container
- ◆ Queda não transparente ao cliente → exceção ao chamar método
- ◆ Criação de um bean entidade → inserção de novo registro na BD
- ◆ Tipos de Persistência: Gerenciada pelo container e Gerenciada pelo componente

Session Beans

- ◆ Representa uma sessão de um único cliente
- ◆ Não representa dados na BD, mas pode atualizá-los
- ◆ Vida relativamente curta
- ◆ Removido quando ocorre queda do container → cliente tem de restabelecer um novo objeto sessão para continuar o trabalho
- ◆ Úteis para descrever e gerenciar interações entre os componentes entidades e implementar um fluxo de trabalho, tipos:
 - Sem estado
 - não mantém o estado entre as chamadas dos métodos
 - Com estado
 - dedicado a um cliente pelo tempo de vida da instância
 - mantém estado conversacional: pode guardar dados relativos ao cliente nas chamadas de métodos

Visão do Cliente

- ◆ *Componente Session* → objeto não persistente em execução em um servidor
- ◆ *Componente Entity* → objeto que fornece uma visão OO de entidades armazenadas num BD, ou de entidades persistentes implementadas por aplicação corporativa já existente
- ◆ *Container* → gera ou fornece as classes que implementam as interfaces Home e Remote
- ◆ *Cliente* → utiliza JNDI para localizar e obter uma referência para interface Home
- ◆ *Através da Home* → o cliente obtém uma referência para interface Remote

Java Enterprise Edition (JEE)

- O que é JEE?
 - É uma plataforma de desenvolvimento de aplicações que faz parte da plataforma Java.
 - Voltada para aplicações multi-camadas, baseadas em componentes que são executados em um servidor de aplicações.
 - É um padrão de desenvolvimento, pois o fornecedor de software deve seguir determinadas regras se quiser declarar os seus produtos como compatíveis com JEE.
 - Ela contém bibliotecas desenvolvidas para o acesso a base de dados, RPC, CORBA, etc.
 - Utilizada principalmente para o desenvolvimento de aplicações corporativas.
-

Java Enterprise Edition (JEE)

- O que é JEE? (cont.)
 - Composta por especificações com funcionalidades distintas:
 - JDBC (Java Database Connectivity), utilizado no acesso a bancos de dados;
 - Servlets, são utilizados para o desenvolvimento de aplicações Web com conteúdo dinâmico;
 - JSP (Java Server Pages), uma especialização do Servlet permitindo que conteúdo dinâmico seja facilmente desenvolvido;
 - JSF (Java Server Faces), framework para simplificar o desenvolvimento de aplicações WEB, baseado em componentes e orientado a eventos;
-

Java Enterprise Edition (JEE)

- O que é JEE? (cont.)
 - JTA (*Java Transaction API*), é uma API que padroniza o tratamento de transações dentro de uma aplicação Java;
 - EJBs (*Enterprise Java Beans*), utilizados no desenvolvimento de componentes de software.
 - Permitem que o programador se concentre nas necessidades do negócio do cliente, enquanto questões de infraestrutura, segurança, disponibilidade e escalabilidade são responsabilidade do servidor de aplicações;
 - JCA (*Java Connector Architecture*), é uma API que padroniza a ligação a aplicações legadas;
 - JPA (*Java Persistence API*), é uma API que padroniza o acesso a banco de dados através de mapeamento Objeto/Relacional dos Enterprise Java Bean;
-

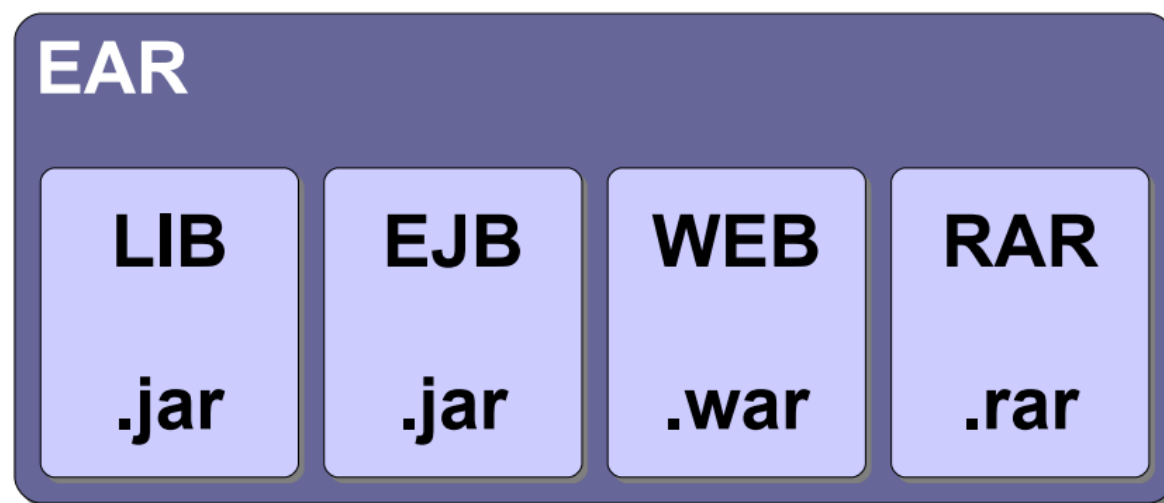
Java Enterprise Edition (JEE)

- Estrutura do JEE
 - Composta por módulos.
 - Cada módulo deve ser compactado no formato ZIP com extensões distintas para ser instalado.

TIPO	EXT.	NOME	DESCRIÇÃO
LIB	.jar	Java Archive	Bibliotecas de classes que são dependências dos outros módulos.
EJB	.jar	Enterprise Java Beans	Componentes SessionBeans, MessageBeans, EntityBeans, etc.
WEB	.war	Web Archive	Aplicação WEB
RAR	.rar	Resource Adapter Archive	Conectores JCA para acesso a sistemas legados.

Java Enterprise Edition (JEE)

- Estrutura do JEE (cont.)
 - Diversos módulos podem ser agrupados em um único arquivo instalável, o EAR.



APIs Java Java Enterprise Edition (JEE) no EJB

API	DESCRIÇÃO	API	DESCRIÇÃO
EJB	define modelo de componente servidor que fornece portabilidade e implementa serviços	Servlets e JSP	Java Servlets e Java Server Pages suporta, HTML dinâmico e gerenciamento de sessão p/ clientes usando browser
JNDI	fornece acesso ao servidor de nomes e diretórios	JMS	Java Messaging Service suporta comunicação assíncrona através de vários sistemas de <i>messaging</i>
RMI	cria interfaces remotas p/ computação distribuída na plataforma JAVA	JTA	Java Transaction API fornece demarcação de transação
JAVA IDL	cria interfaces remotas p/ suportar comunicação CORBA. Inclui compilador IDL e um ORB enxuto que suporta IIOP	JTS	Java Transaction Service define um serviço de gerenciamento de transações distribuídas baseado no OTS CORBA
JDBC	fornece acesso uniforme a bancos de dados relacionais		