

Aplicação da Arquitetura Multicamadas Utilizando Java

Raquel Schlickmann

Orientador: Marcel Hugo

Roteiro

- Introdução
- Histórico da Arquitetura de Software
- Enterprise JavaBeans - EJB
- Voyager
- Implementação
- Conclusões

Introdução

- Justificativa
 - Aplicações distribuídas :
 - Arquitetura Multicamadas - 3 camadas lógicas
 - Apresentação
 - Negócio
 - Acesso a dados
- Objetivo

Estudar e aplicar a arquitetura multicamadas utilizando a linguagem java.

Histórico da Arquitetura de Software

- Troca de Paradigmas
- Arquitetura de uma Camada
- Arquitetura de duas Camadas
- Arquitetura Multicamadas

Troca de Paradigmas

- Primeira “onda” - aplicações monolíticas
- Segunda “onda” - aplicações cliente/servidor
- Terceira “onda” - aplicações distribuídas

Histórico da Arquitetura de Software

Troca de Paradigmas

- Arquitetura de uma Camada
- Arquitetura de duas Camadas
- Arquitetura Multicamadas

Arquitetura de uma Camada

- Vantagens
 - fácil de gerenciar e controlar
 - seguro
 - confiável
 - suporta vários usuários
- Desvantagens
 - escalabilidade
 - dependência de hardware e software

Histórico da Arquitetura de Software

 Troca de Paradigmas

 Arquitetura de uma Camada

- Arquitetura de duas Camadas
- Arquitetura Multicamadas

Arquitetura de duas Camadas

- Vantagens
 - ferramentas GUI (*Graphical User Interface*)
 - custos mais baixos que *mainframes*
 - banco de dados independentes de plataformas
- Desvantagens
 - perda de segurança
 - perda de confiança
 - perda de controle

Histórico da Arquitetura de Software

 Troca de Paradigmas

 Arquitetura de uma Camada

 Arquitetura de duas Camadas

- Arquitetura Multicamadas

Arquitetura Multicamadas



Arquitetura Multicamadas

- Características
 - escalabilidade e performance
 - suporte a sistemas críticos
 - gerenciabilidade
 - flexibilidade
 - reusabilidade e integração
 - suporte a multi-clientes

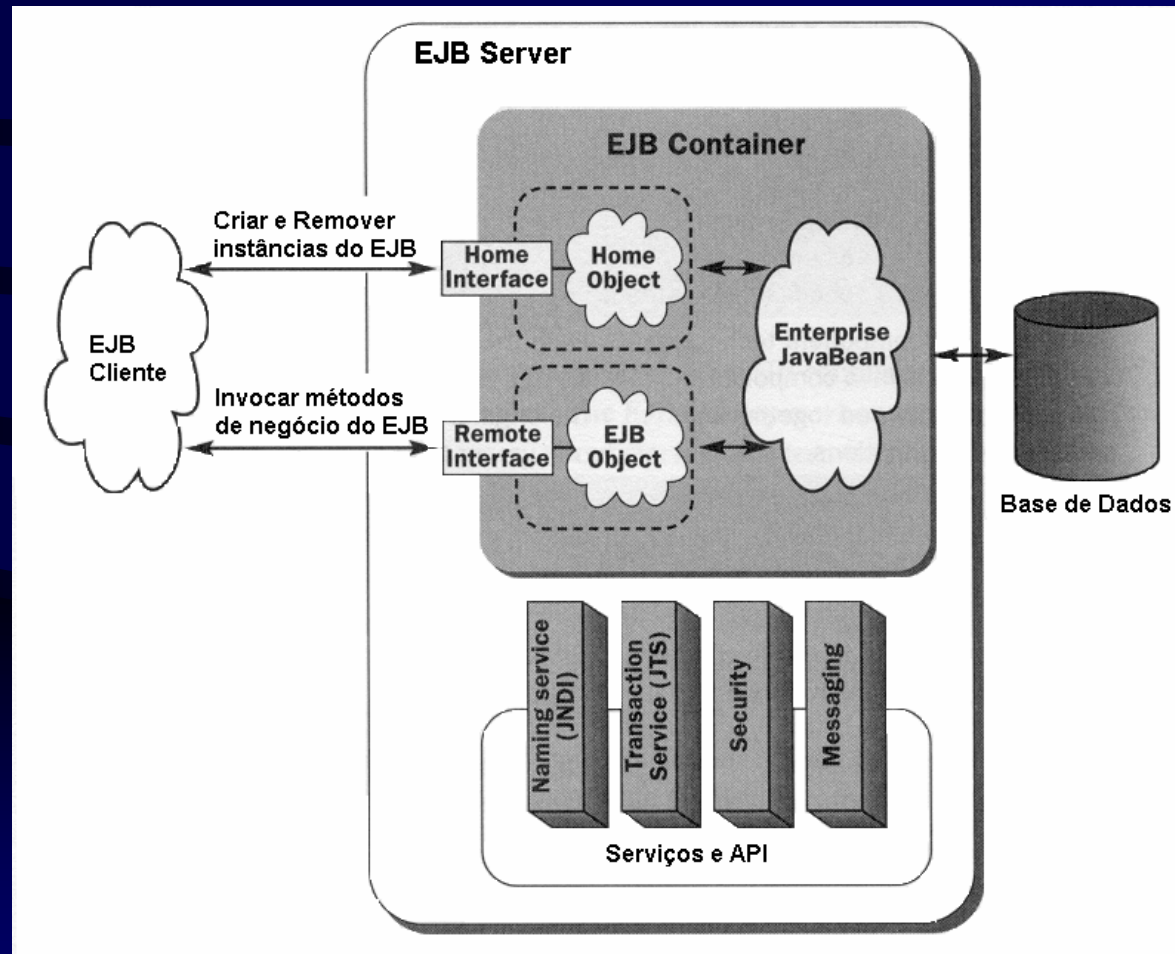
Arquitetura Multicamadas

- Vantagens
 - desenvolvimento mais rápido
 - reutilização de objetos por outras aplicações
 - facilidade de manutenção
 - independência de fornecedor de banco de dados
 - abstração da lógica de acesso a dados
 - alta produtividade
 - infra-estrutura distribuída de computação

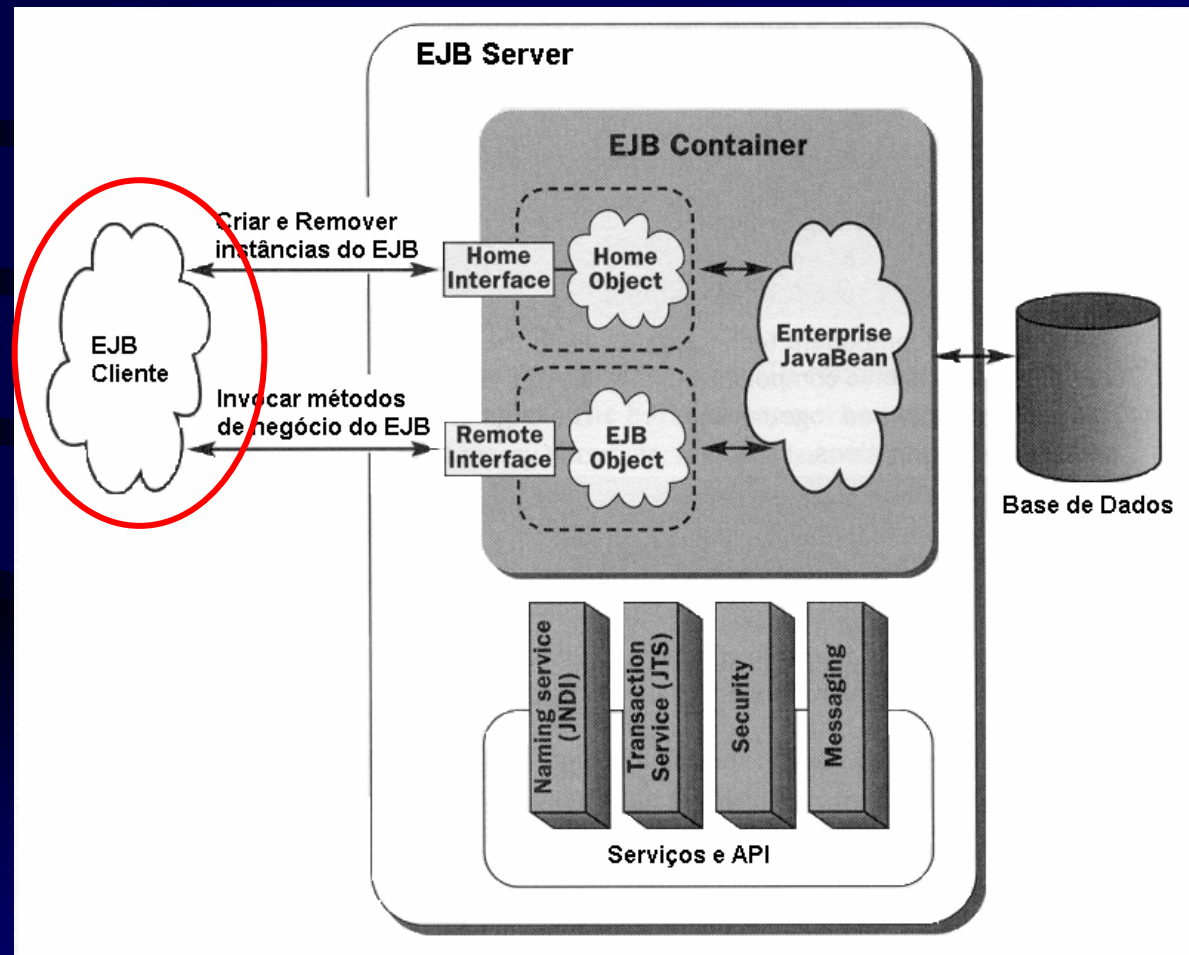
Histórico da Arquitetura de Software

- ☞ Troca de Paradigmas
- ☞ Arquitetura de uma Camada
- ☞ Arquitetura de duas Camadas
- ☞ Arquitetura Multicamadas

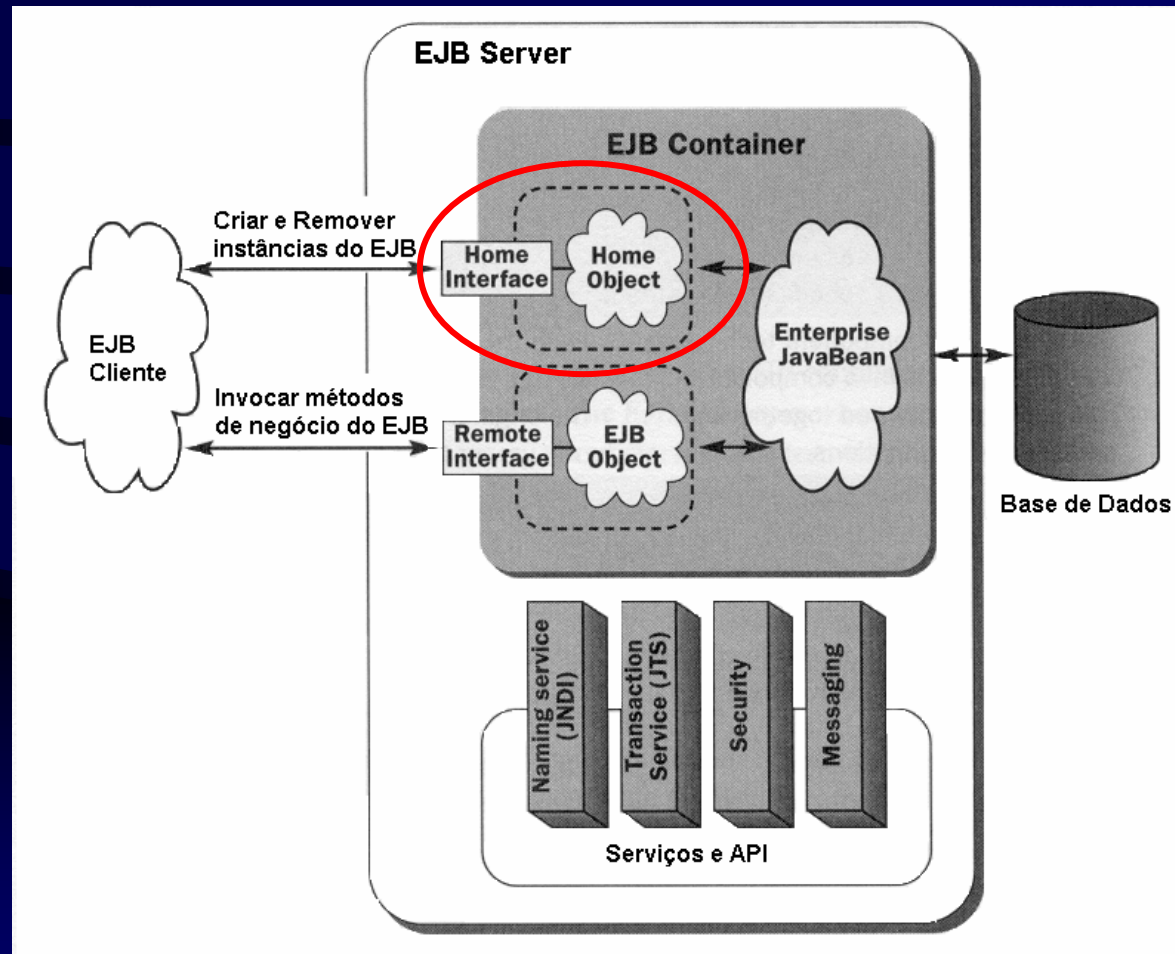
Enterprise JavaBeans - EJB



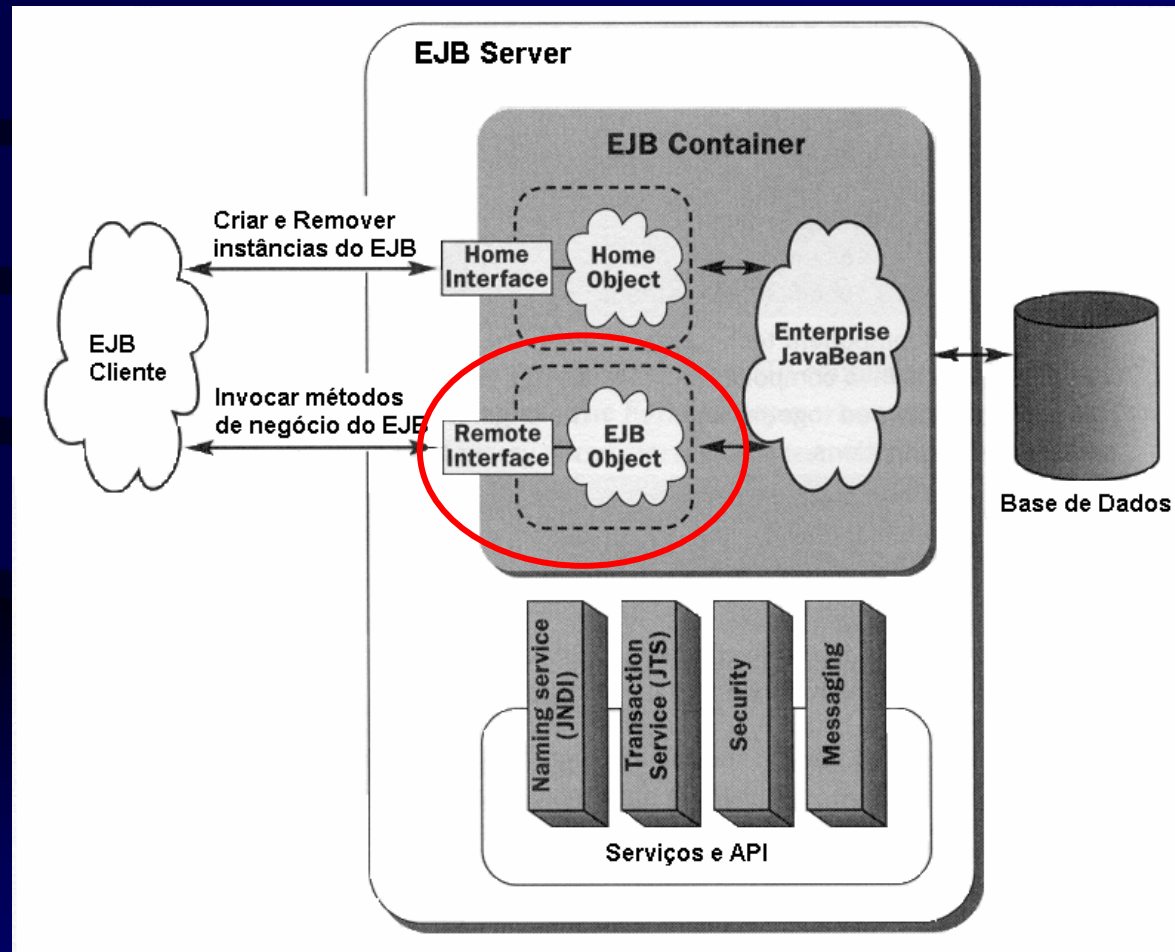
EJB Cliente



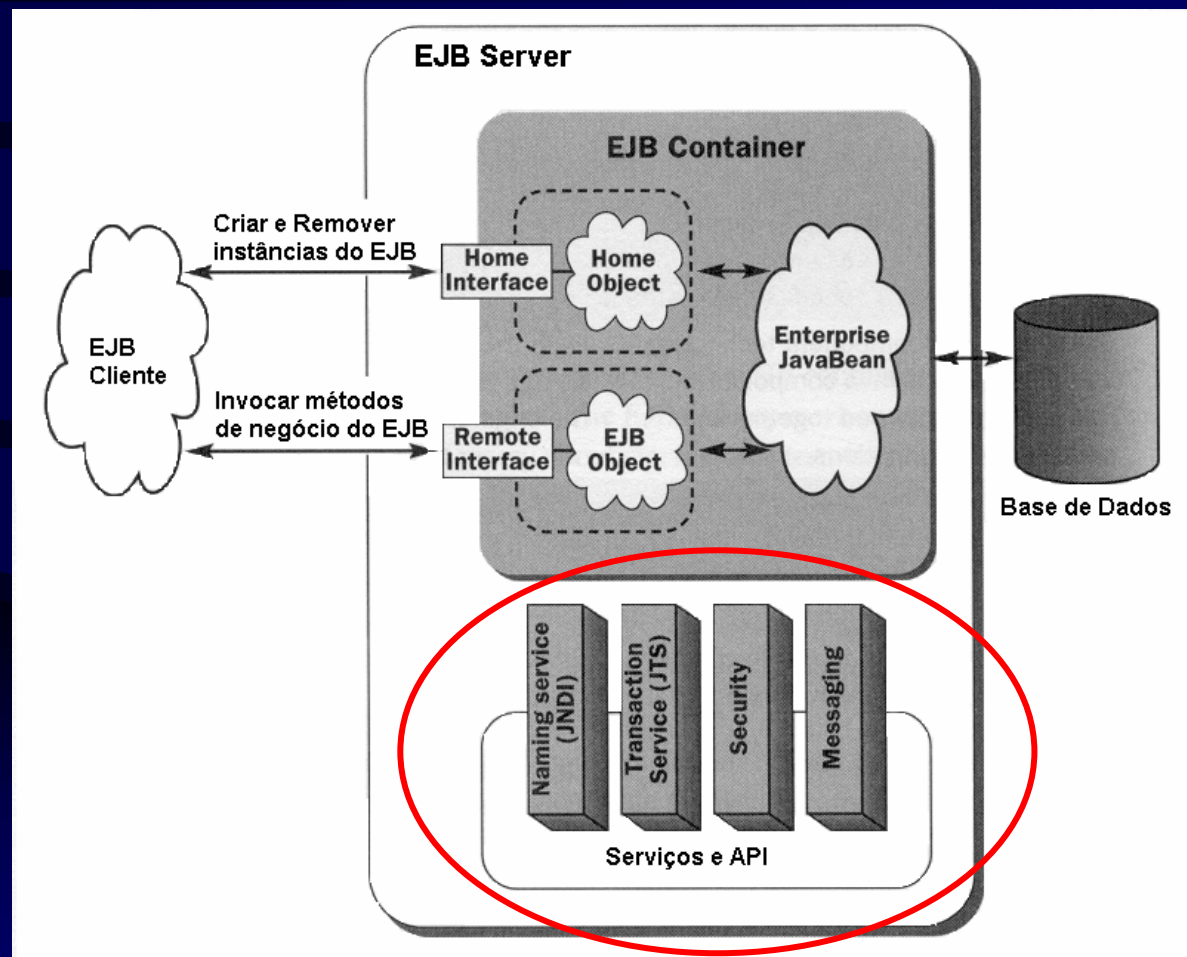
Home Interface



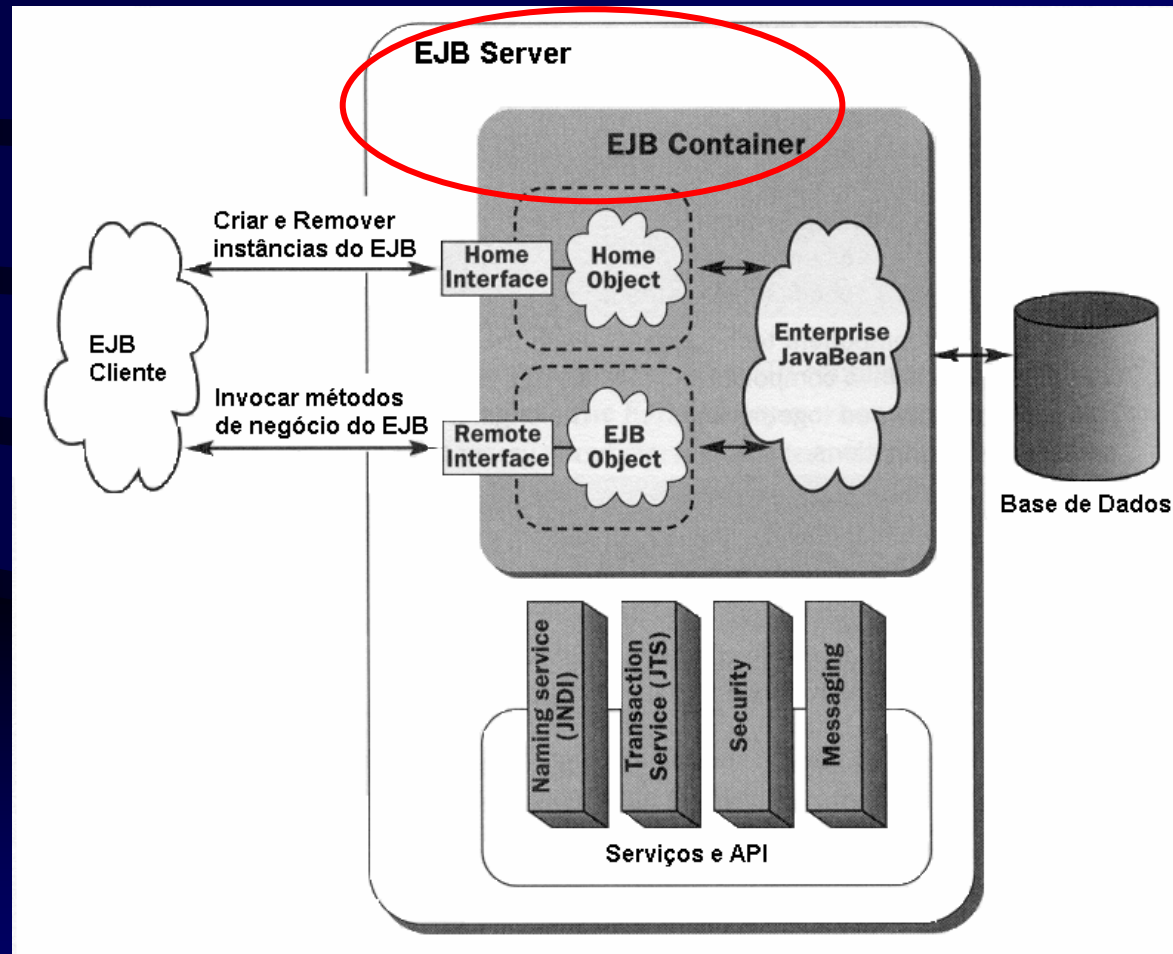
Remote Interface



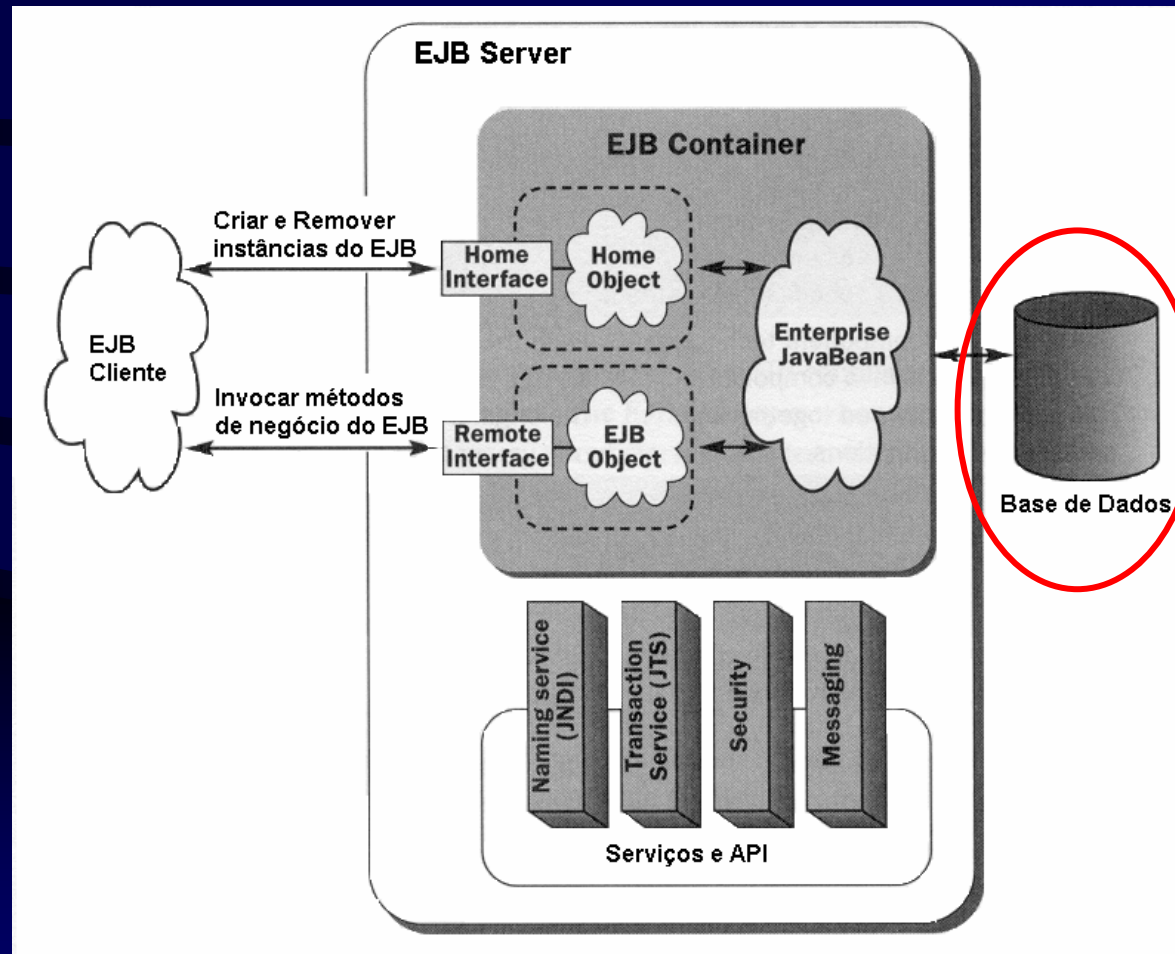
Serviços e API



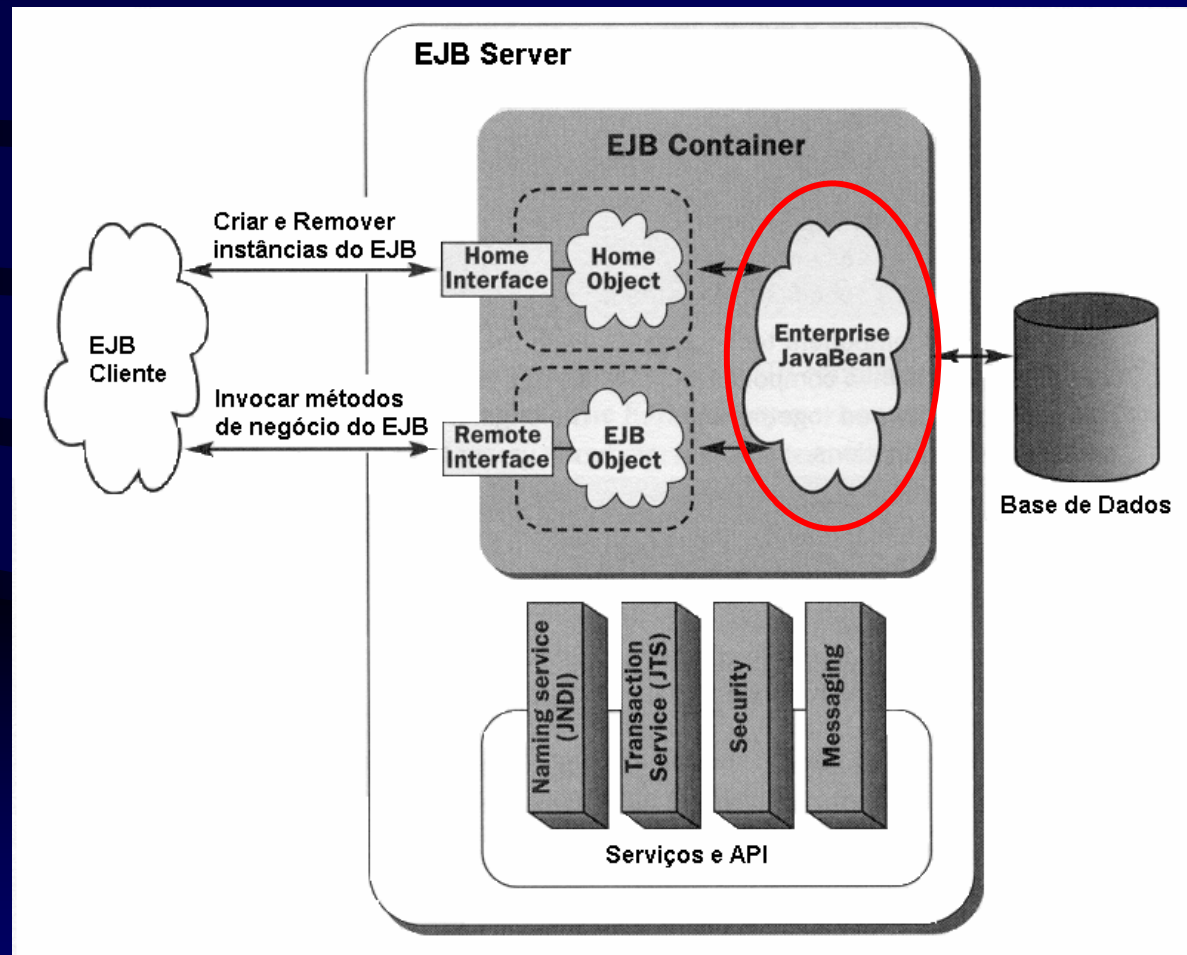
EJB Server e EJB Container



Base de Dados



Enterprise JavaBeans



Tipos de Enterprise JavaBeans

- *Entity Beans*
- *Session Beans*
 - *Stateless Session Beans*
 - *Stateful Session Beans*

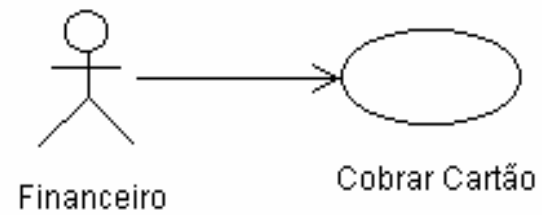
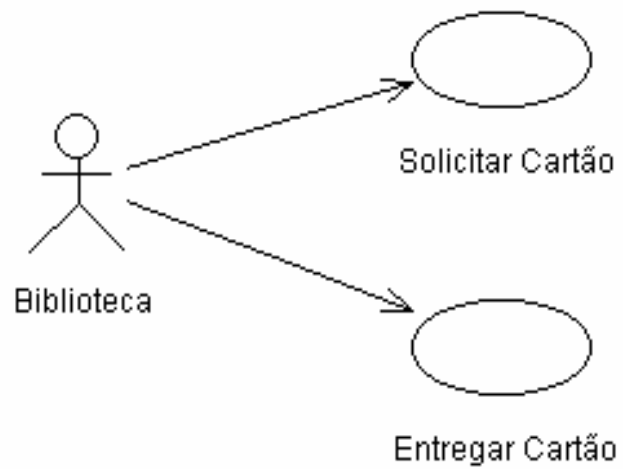
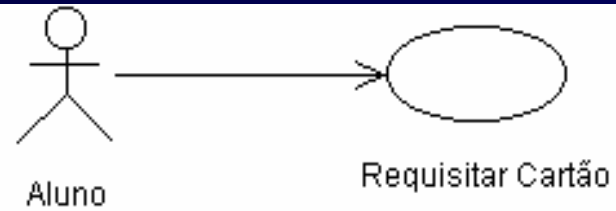
Voyager

- Servidor de aplicação
- Container

Implementação

- Cenário
- Especificação
- Exemplo de Entity Beans
- Arquitetura do Protótipo

Cenário

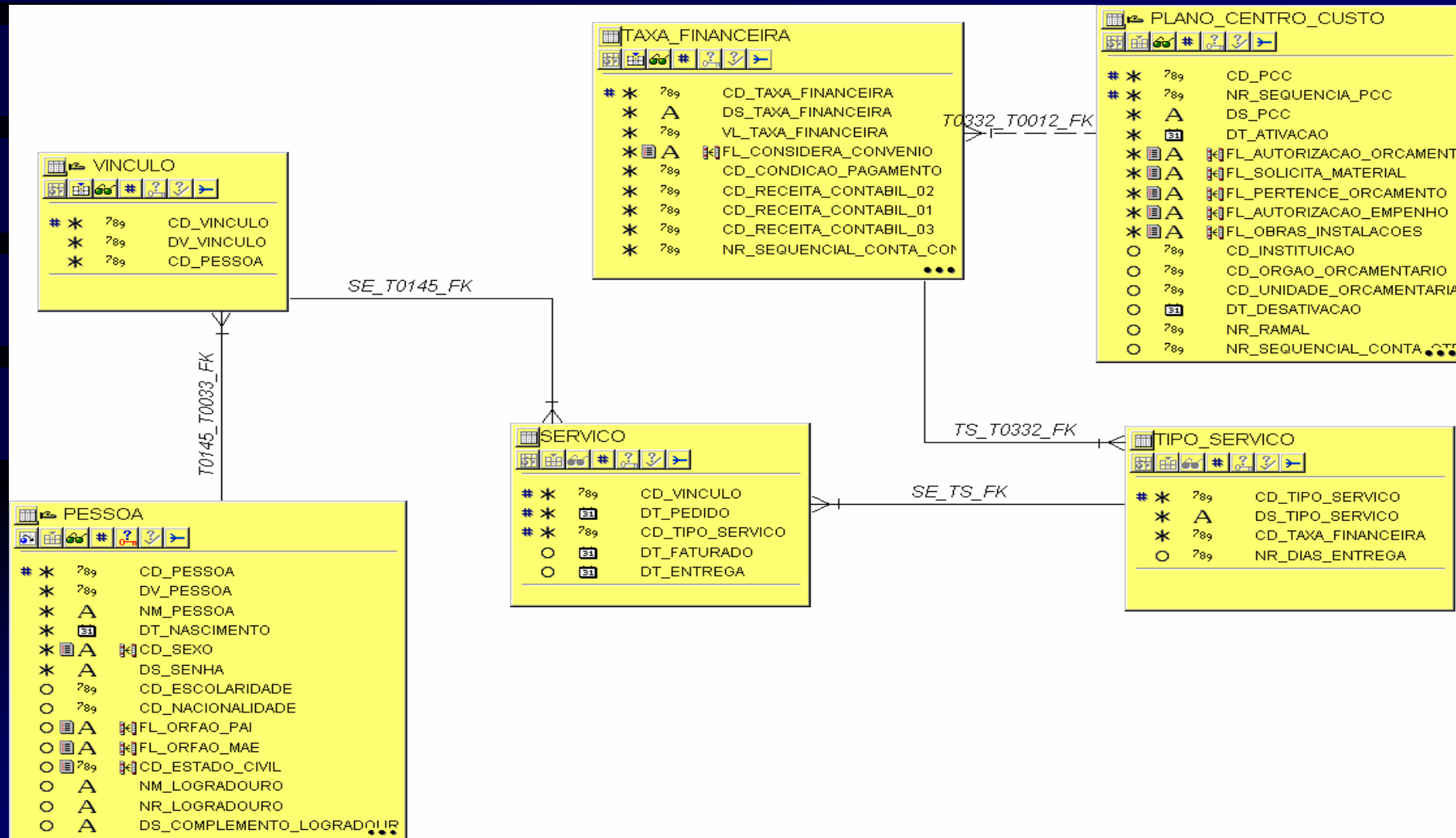


Implementação

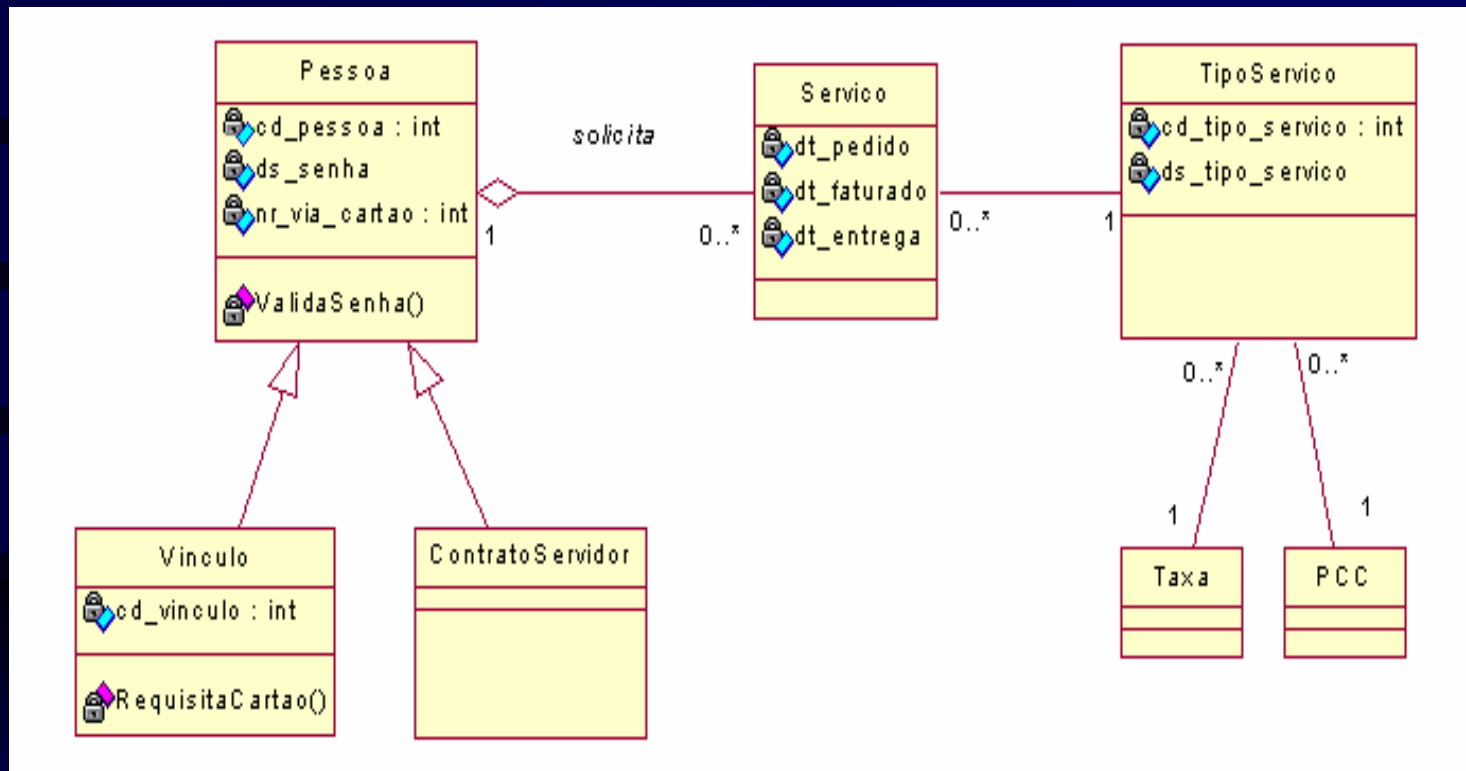
Cenário

- Especificação
- Exemplo de Entity Beans
- Arquitetura do Protótipo

Especificação Estruturada



Especificação Orientada a Objetos



Implementação

 Cenário

 Especificação

- Exemplo de Entity Beans
- Arquitetura do Protótipo

Exemplo de um Entity Bean

Home Interface da Classe Pessoa

```
package tcc;

import javax.ejb.*;
import java.rmi.*;
import java.util.Enumeration;

public interface IPessoaHome extends EJBHome {
    IPessoa create(int cd_pessoa) throws RemoteException,
    CreateException;
    IPessoa findByPrimaryKey(int cd_pessoa) throws RemoteException,
    FinderException;
    Enumeration findByCodigoId(int cd_pessoa) throws FinderException,
    RemoteException;
}
```

Exemplo de um Entity Bean

Remote Interface da Classe

Pessoa

```
package tcc;
import javax.ejb.*;
import java.rmi.*;

public interface IPessoa extends EJBObject {
    public boolean ValidaSenha(String ds_senha) throws RemoteException;
    public void AlteraPessoa() throws RemoteException;
}
```


Implementação

 Cenário

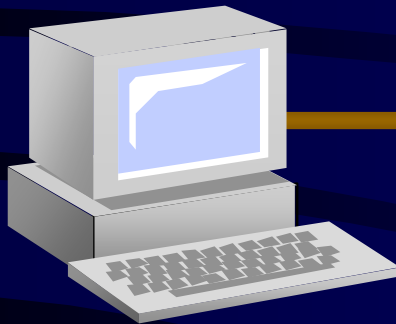
 Especificação

 Exemplo de Entity Beans

- Arquitetura do Protótipo

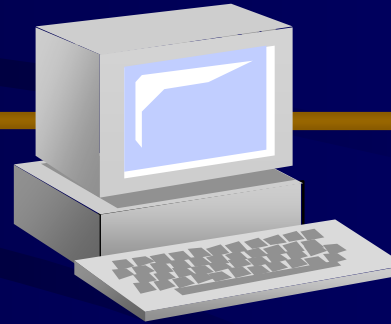
Arquitetura do Protótipo

Cliente



Java
PC - Windows 95

Servidor de Aplicação



Voyager
PC - Windows 95

Servidor de Dados



ORACLE 8.0.5
Risc RS6000
UNIX AIX 4.3.1

Implementação

 Cenário

 Especificação

 Exemplo de Entity Beans

 Arquitetura do Protótipo

Conclusões

- Multicamadas:
 - clientes “leves”
 - facilidade de manutenção
 - reutilização
- Enterprise JavaBeans
 - componentes reutilizáveis
 - facilidade da implementação desta arquitetura