Desenvolvimento de Sistemas com Frameworks

UniMB
CENTRO UNIVERSITÁRIO MONTES BELOS

Curso: Analise e Desenvolvimento de Sistemas
Turma: TAS05-NA
Prof.: Wender da Silva Paula

Agenda

- Introdução ao Java EE (Java Enterprise Edition)

Java EE (Java Enterprise Edition)

O **Java EE** (Java Enterprise Edition) consiste de uma série de especificações bem detalhadas, dando uma receita de como deve ser implementado um software que faz cada um desses serviços de infraestrutura.

É uma plataforma ou ambiente para desenvolvimento de aplicações de grande porte e aplicações web que possui bibliotecas e funcionalidades que implementam softwares baseados na linguagem Java.

A plataforma Java EE oferece ao cliente e ao desenvolvedor recursos para as criações de sistema com segurança, escalabilidade, integridade, confiabilidade entre outros.

Java EE (Java Enterprise Edition)

Essa plataforma tem uma série de tecnologias que têm objetivos distintos, sendo que as mais conhecidas são:

- **Servlets** – São componentes Java executados no servidor que tem o objetivo de gerar conteúdo (HTML e XML) dinâmico para web.
- **Java Server Pages (JSP)** – Especialização de servlets que permite aplicações em java serem mais robustas e tenham facilidade no desenvolvimento.
- **Java Serve Faces (JSF)** – É framework web com o padrão **MVC** (Model, View e Controller) baseado em Java que ajuda a simplificar o desenvolvimento da interfaces (telas do sistema) através de um modelo de UI.

Java EE (Java Enterprise Edition)

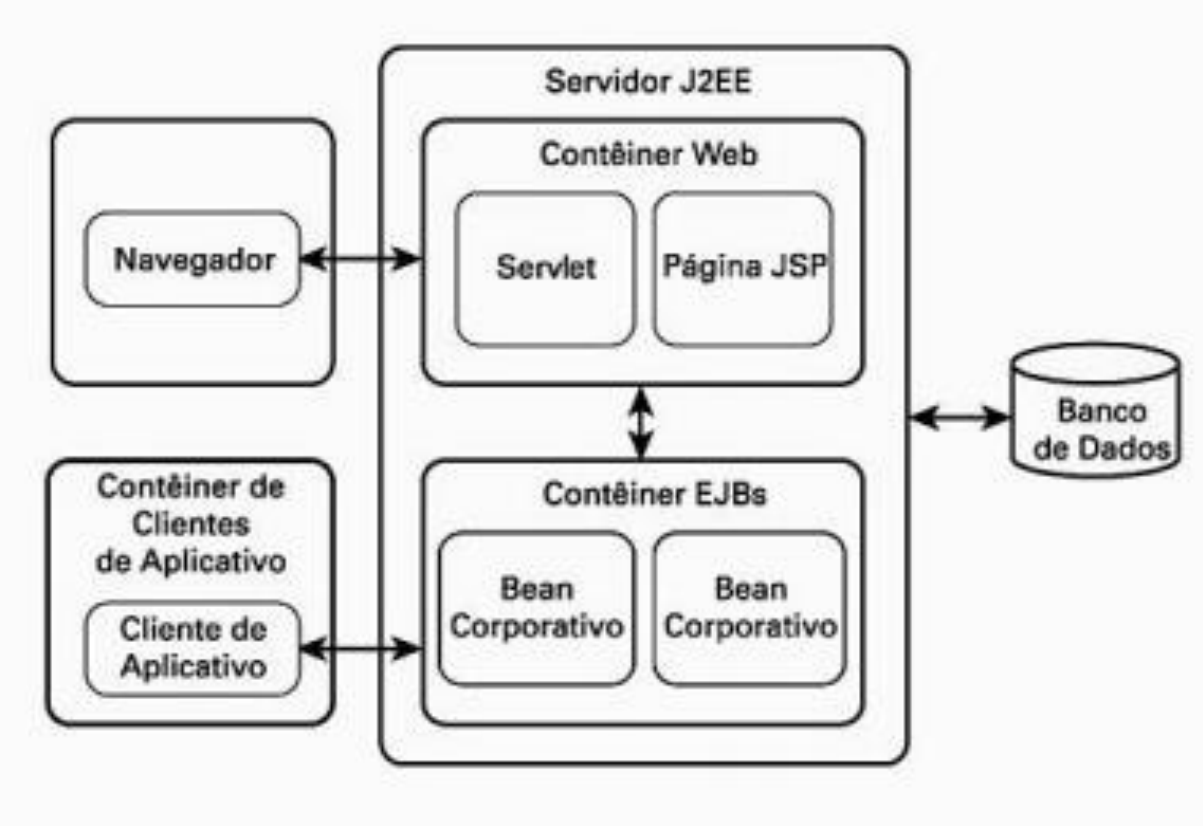
- **Java Persistence API (JPA)** – É uma API padrão do Java que usa o conceito de mapeamento objeto relacional, sendo utilizada para a persistência dos dados. Essa ferramenta traz muita produtividade, pois consegue desenvolver aplicações que trabalham com banco de dados sem escrever nenhuma linha SQL.
- **Enterprise java Beans (EJB)** – São componentes que executam em um container de aplicação e que oferecem a facilidade e produtividade no desenvolvimento de componentes distribuídos, transacionados, seguros e portáteis.

Contêineres e Componentes

Para fornecer a comunicação entre componentes, APIs, persistência, serviços entre outros, o Java EE oferece os contêineres para abrigá-los. Os contêineres são implementados pelos fornecedores de servidor de aplicativos Java EE, que disponibilizam um container para cada tipo de componente como: Applet, cliente de aplicativo, web e EJBs.

Contêineres e Componentes

Os componentes Web e EJB são distribuídos, gerenciados e executados em um servidor Java EE. Veja como funciona na Figura **Arquitetura Lógica do Java EE**.



Contêineres

Os contêineres armazenam, oferecem serviços (gerenciamento ciclo de vida, segurança, implementação, e comunicação com outros componentes) e recursos como um ambiente em tempo de execução compatível para os **componentes Java EE conseguirem trabalhar**.

Serviços dos Contêineres

Os contêineres fornecem a cada tipo de componente um conjunto de serviços:

- **Conectividade**: o tipo de conectividade atribuída é realizada pelos objetos distribuídos através de RMI (Remote Method Invocation) e CORBA. Porém, todo o ciclo da conectidade deve ser fornecido através dos protocolos HTTP e HTTPS.
- **Serviços de Diretório**: Conhecido como JNDI (Java Naming and Directory Interfaces), è uma API que acessa serviço de diretórios, sendo que cada serviço deve fornecer um provedor de serviços (Service Provider). Os servidores Java EE são obrigado a possuir esse tipo de serviço.

Serviços dos Contêineres

- **JDBC:** O acesso aos dados acontece através da API JDBC (Java DataBase Connection) que permite conectar com um banco de dados através de um driver específico para o tipo de banco (Mysql, Oracle, Postgresql e outros). Com essa API também consegue-se executar instruções SQL que auxiliam na manipulação das informações.
- **JCA (Java Connector Architecture):** Conhecido também como conexão legada, ajuda a fornecer suporte de integrações de servidores de informações corporativas e sistemas legados. Esse componente ajuda muito no processamento de transação de ERPs e computadores de grande porte.
- **Segurança:** Usa APIs como **JASS (Java Authentication and Authorization Service)** que é um serviço que ajuda na autenticação e autorização dos usuários no sistema de plataformas java EE.

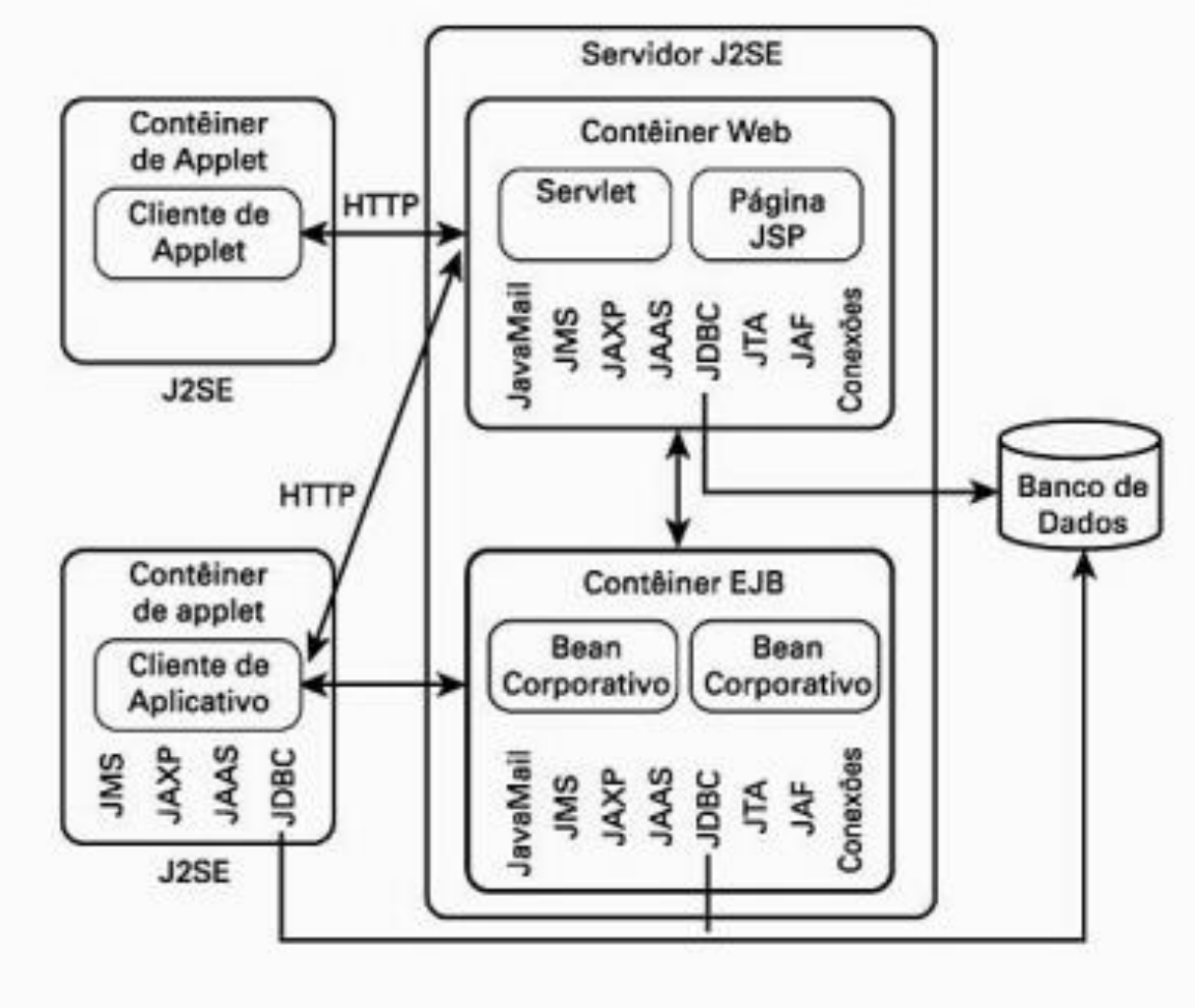
Serviços dos Contêineres

- **XML**: Permite suportar XML (eXtensible Markup Language) usando **DOM** (Document Object Model), documentos **SAX** (Simple API for XML) e XSLT (extensible Stylesheet Language Transformations).
- **Transações**: Em um servidor Java EE é fornecido serviços de transações para todos seus componentes. Geralmente, o container assume essa responsabilidade, porém a **JTA** (Java Transaction API) permite que o componente controle suas próprias transações.
- **JTA (Java Transaction API)**: É uma API que permite trabalhar com transações independentes do gerenciador.

Serviços dos Contêineres

- **E-mail**: Esse componente permite a troca de mensagens através do serviço de mensagens do Java, conhecido como **JMS (Java Message Service)**. Nesse serviço existe a **API JavaMail** que oferece manipular os envios e recebimentos de e-mail e troca de mensagens entre clientes. Especificamente, o JavaMail permite enviar e receber e-mail utilizando vários protocolos como: POP, SMTP e IMAP.
- **JMS (Java Message Service):** É uma API que oferece o mecanismo de criar, ler e armazenar mensagens.

Os principais serviços do Java EE



Conhecendo os componentes no Java EE

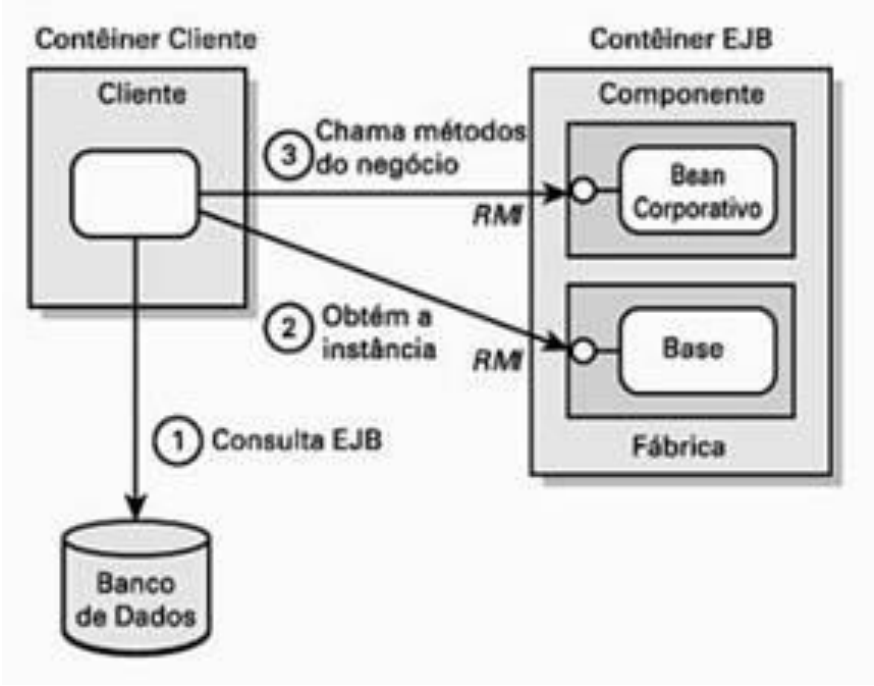
Os componentes são divididos em **EJBs** e **Componente Web**. Os **EJBs** possuem outras subdivisões: **Beans de sessão**, **Beans de Mensagem** e **Beans de Entidade**.

- **Enterprise JavaBeans – EJB**

Os EJBs são responsáveis por facilitar o encapsulamento e compartilhamento da lógica do negócio. Para expor a lógica de negócio para outros componentes conseguirem utilizar e facilitar a interação entre o EJB e o cliente é envolvido dois mecanismos: RMI (Invocação de métodos remotos) e a JNDI (Interface de atribuição de nomes e diretórios do Java).

Conhecendo os componentes no Java EE

Uso de um cliente com a JNDI e RMI para acessar um EJB



Conforme mostrado na **Figura**, um cliente usa a JNDI para procurar a localização do EJB, interagindo com a Fábrica (Base do EJB) para obter uma instância do EJB.

Conhecendo os componentes no Java EE

Enterprise JavaBeans – EJB

- **Beans de Sessão**: Esse tipo permite que a regra de negócio seja desenvolvida e depois implantada, independentemente da camada lógica de apresentação, ou seja, é uma extensão da funcionalidade do negócio de um cliente na camada intermediária. Um exemplo prático é quando há necessidade de desenvolver certas regras no login do sistema para tipos de usuários.
- **Beans de Entidade**: Esses beans representam um dado corporativo (entidade) durante os processos, sendo possível com que o cliente consiga acessar os dados e funcionalidade. Podem também conseguirem acessar banco dados ou sistemas de ERP, com o objetivo para reunir todas as informações corporativas que representam.
- **Beans de Mensagem**: Esse tipo é associado a uma fila de mensagens em particular que armazena todas as mensagens que chegam nessa fila, sendo distribuídas para uma instância do bean dirigido por mensagens. O objetivo é abrigar a lógica de negócio, em vez de dados, acessando os dados exigidos através do JDBC ou beans de entidade.

Conhecendo os componentes no Java EE

Enterprise JavaBeans – EJB

Observe a figura: EJBs separando a lógica do negócio em lógico do aplicativo e do domínio

Conhecendo os componentes no Java EE

Algumas Vantagens dos Componentes Corporativos

- **Eficiência**: Existem componentes que oferecem a agilidade no desenvolvimento nas telas de apresentação do sistema conhecidas como GUIs (interfaces do usuário) e com isso ajuda a identificarem a lógica de negócio e acesso aos dados.

- **Extensibilidade**: permite adicionar ou remover componentes em um aplicativo para oferecer mais funcionalidade.

- **Independência**: Permite ter a independência de linguagens.

- **Atualização do sistema:** O uso de componente permite a alteração sem afetar outros componentes do sistema.

Conhecendo os componentes no Java EE

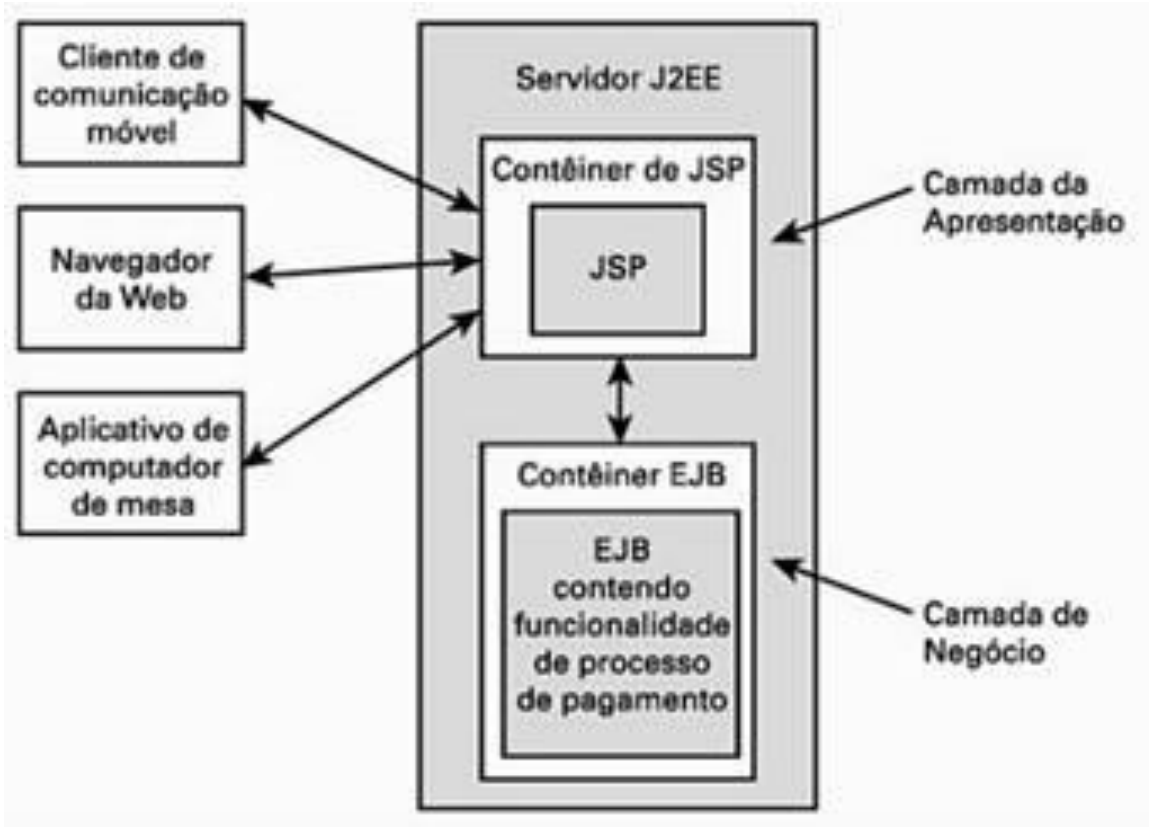
Camadas

Na plataforma Java EE existem três tipos de camadas: **Camada de Negócio**, **Camada de Apresentação** e **Camada Cliente**.

- **Camada de Negócio:** Essa camada também pode ser conhecida como camada física e tem o objetivo de encapsular a lógica do negócio ou EJBs que são usados pelos componentes da camada da apresentação que fornecem essa funcionalidade para usuários do aplicativo.

Conhecendo os componentes no Java EE

Simulação de acessos simultâneos.



Conhecendo os componentes no Java EE

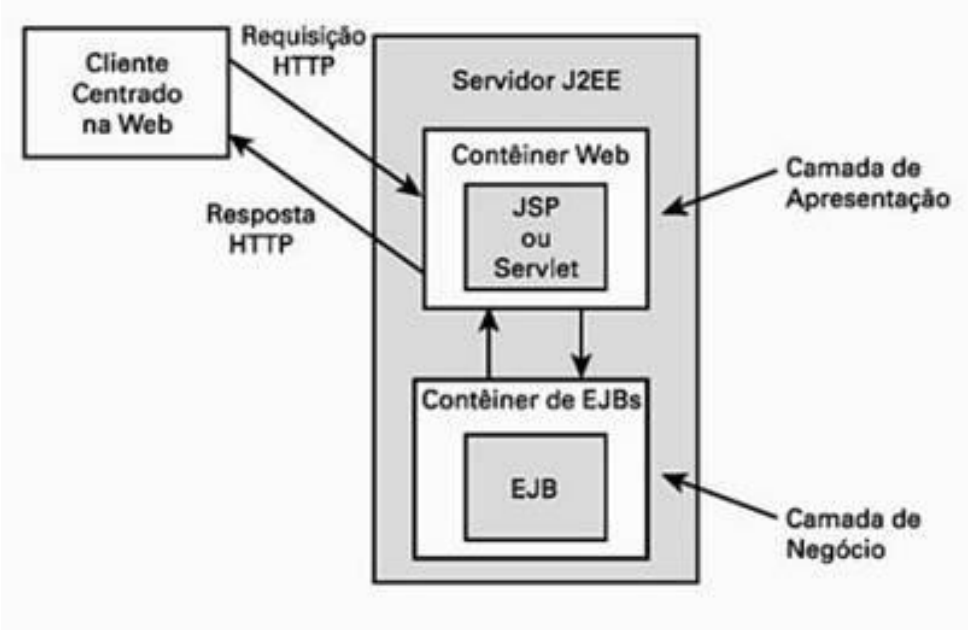
Camadas

- **Camada de Apresentação:** Essa camada é considerada como lógica de apresentação, pois governa as telas que são exibidas para o usuário e de como a lógica interagem entre a regra de negócio na sua camada respectiva para conseguirem trabalhar no processo corporativo. Também se leva em conta a maneira que o usuário visualiza a página, pois tudo depende do tipo de cliente (browser) que está tentando acessar.

- **Componente Web**: Esses componentes oferecem a comunicação entre a camada de apresentação com o usuário que acontece somente por dois componentes oferecidos pelo Java EE: **JSP, servlets e Web Services**

Conhecendo os componentes no Java EE

Exemplo de componentes centrados na Web



Na **Figura** é apresentado um exemplo de requisição do cliente para um servlets ou JSP com dados fornecidos por um EJB. No lado do servidor é analisada a requisição e logo após chama-se o EJB que é responsável por conter toda a lógica de negócio da aplicação. Nesse fluxo, quando a servlet ou JSP recebem uma resposta EJB, passam a ser responsáveis por apresentar os dados que recebem. Logo depois que o servlet ou JSP possuir uma resposta, é devolvida para o cliente que acaba completando o ciclo requisição-resposta.

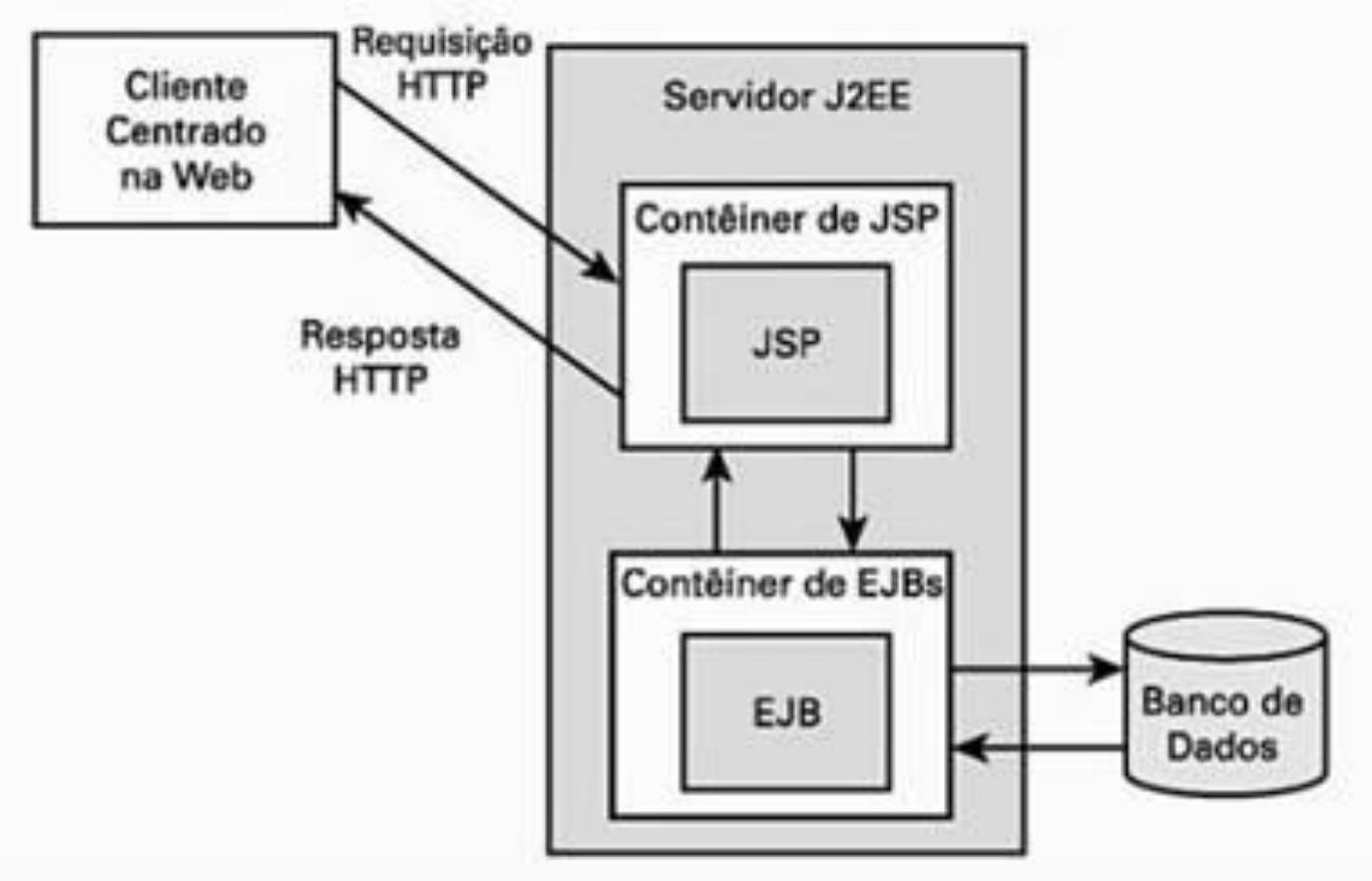
Conhecendo os componentes no Java EE

- **JSP:** Considerada uma tecnologia baseada em Java que permite criar páginas dinamicamente HTML, XML que consegue resposta às requisições dos clientes. Uma das fortes características é o poder de combinar tags JSP e scriptlets que possuem códigos executáveis e marcações estáticas. Uma das vantagens é que esses códigos que ficam armazenados nas páginas JSPs são executados pelo servidor e a página resultante é enviada para o cliente, sendo assim o cliente não visualiza nada de processamento apenas o resultado que retorna na página.

- Na **Figura seguinte** é mostrado como funciona o fluxo de uma requisição HTTP do cliente para uma página JSP. Nesse caso, no momento da requisição, o contêiner de JSPs é manipulado, fazendo a conversão em um servlet. Após isso, o servlet encaminha a requisição para um componente de lógica do negócio, como outra servlet, para um JavaBean ou para um EJB que pode fazer algum processamento de acesso ao banco de dados ou tratamento de alguma regra.

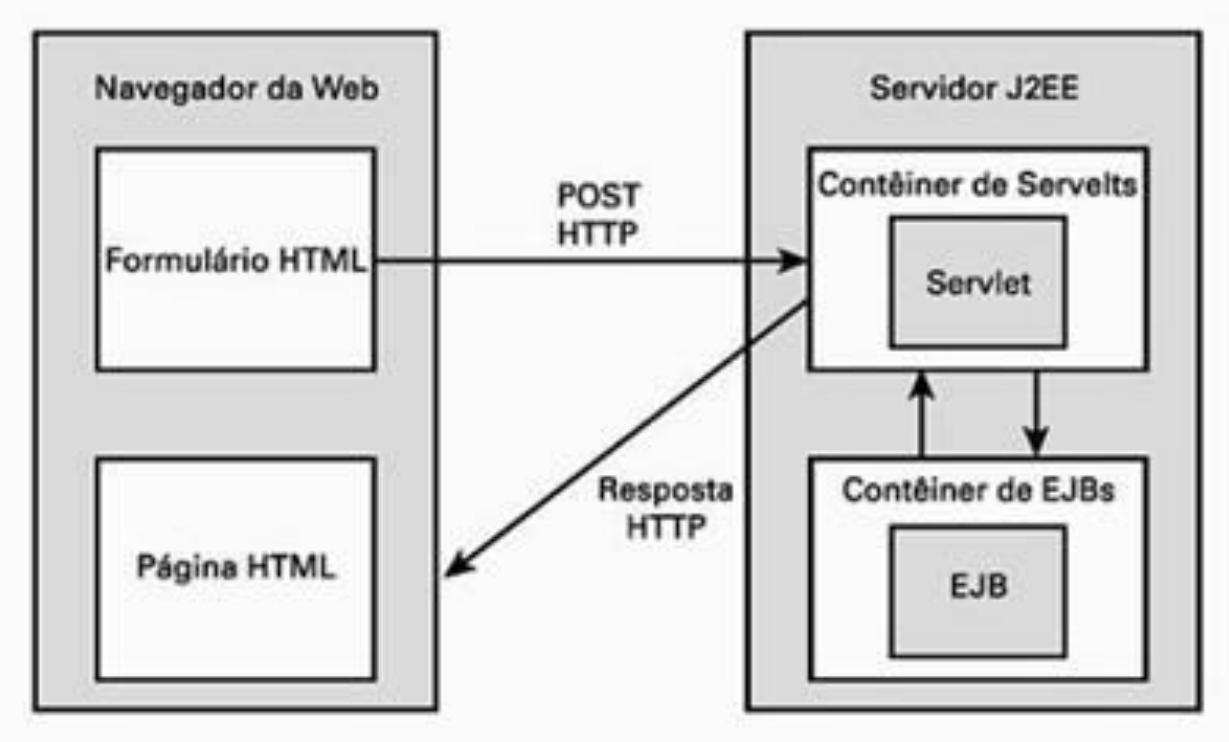
Conhecendo os componentes no Java EE

Fluxo de requisição nas páginas JSPs.



Conhecendo os componentes no Java EE

- **Servlets:** Os servlets são códigos Java que processam requisições (**request**) e resposta (**response**) em servidores que suportam essa tecnologia. Uma das habilidades dos servlets é a facilidade na comunicação com outros componentes Java EE. Observe na f**igura** como funciona o processo de requisição de um cliente para uma servlet e um EJB.



Conhecendo os componentes no Java EE

- **Web Services**: São componentes de middleware baseados em XML (eXtensible Markup Language) que permitem os aplicativos acessarem através de HTTP e SOAP, por isso conseguem consumir serviços web.

- **O middleware** é um programa que faz a interface entre o software e outras aplicações. O motivo da existência é porque pode ser utilizado para mover ou transportar informações e dados entre programas de diferentes protocolos de comunicação, plataformas e dependências do sistema operacional em questão.

Conhecendo os componentes no Java EE

Camada Cliente

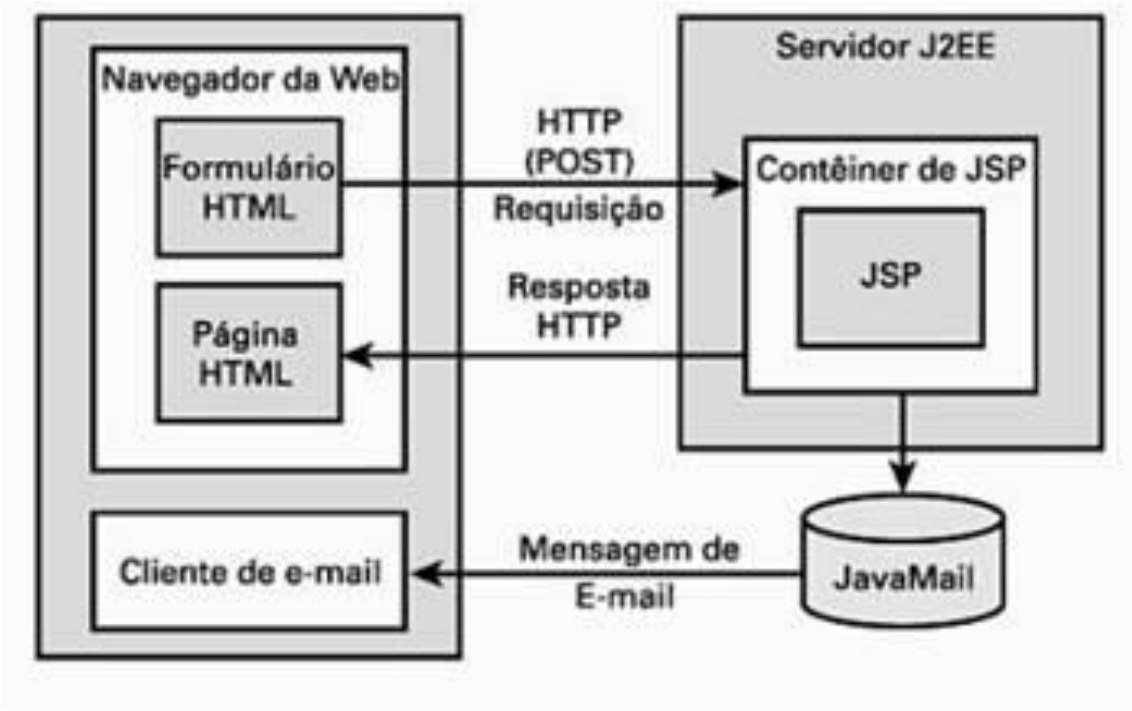
Suporta clientes web e portáteis (dispositivos móveis) que se comunicam através do protocolo HTTP, porém outros conseguem usar SOAP e outros tipos. Essa camada está formada por vários clientes como: HTML estático, HTML dinâmicos, Applet Java, independentes, empresa para empresas, clientes não java e outros clientes HTTP.

- **Cliente - HTML Estático**: Quando a internet surgiu, as primeiras páginas web eram estáticas, ou seja, páginas sem interação, pois os sites possuíam pequenos recursos e poucas funcionalidades como: navegação limitada, processamento básico de formulários de resposta para o usuário e comercialização de e-mails.

Conhecendo os componentes no Java EE

Camada Cliente

- Na figura ao lado é mostrado um exemplo de uma aplicação web no momento em que o usuário preenche os dados de um formulário HTML e efetua uma **requisição via POST** através do servlet correspondente. Após os dados chegarem ao servlet, é analisado e enviado as informações para o componente JavaMail.



Conhecendo os componentes no Java EE

Camada Cliente

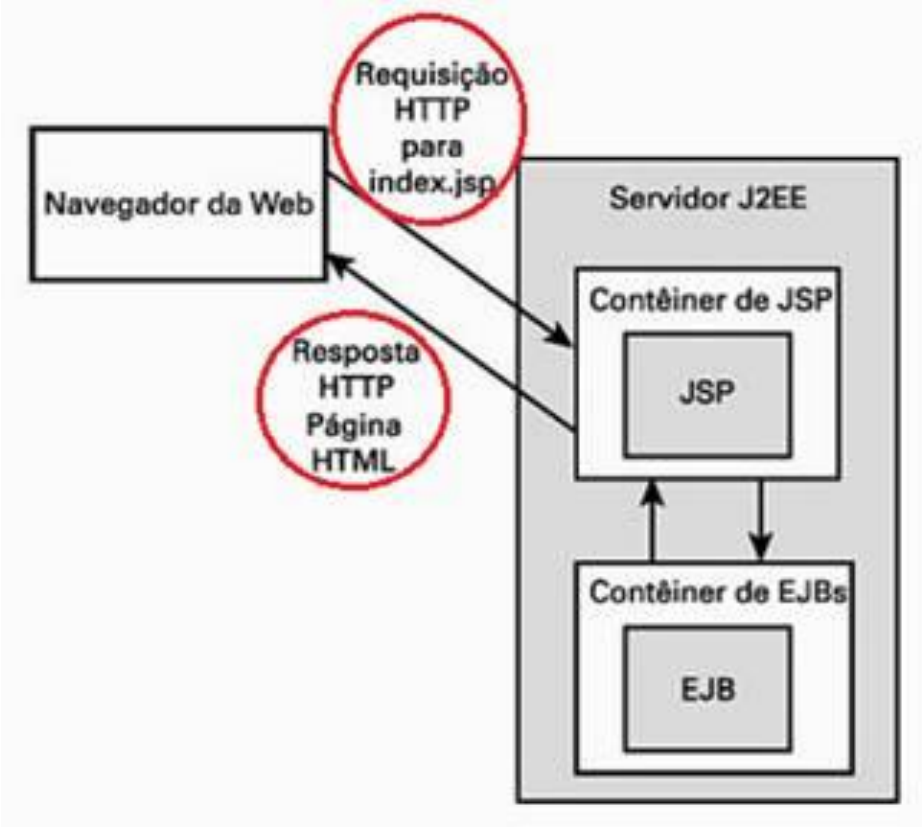
- **Cliente - HTML Dinâmico:** Geralmente, são aplicações web que usam o acesso ao banco de dados que efetuam algum tratamento ou recuperam uma informação que precisa ser armazenada ou exposta para o usuário.

- Na figura seguinte é mostrado um exemplo da **requisição HTTP** para a página inicial (índex.jsp). Pelo fato de que o mecanismo JSP interprete as tags dentro da página, isso acaba chamando o EJB que retorna uma resposta e em seguida é retornado uma página de **código HTML** para o cliente. Nesse exemplo, os **tratamentos de regras** de negócios são encapsulados no EJB.

Conhecendo os componentes no Java EE

Camada Cliente

- Cliente Web interagindo com uma JSP.



Conhecendo os componentes no Java EE

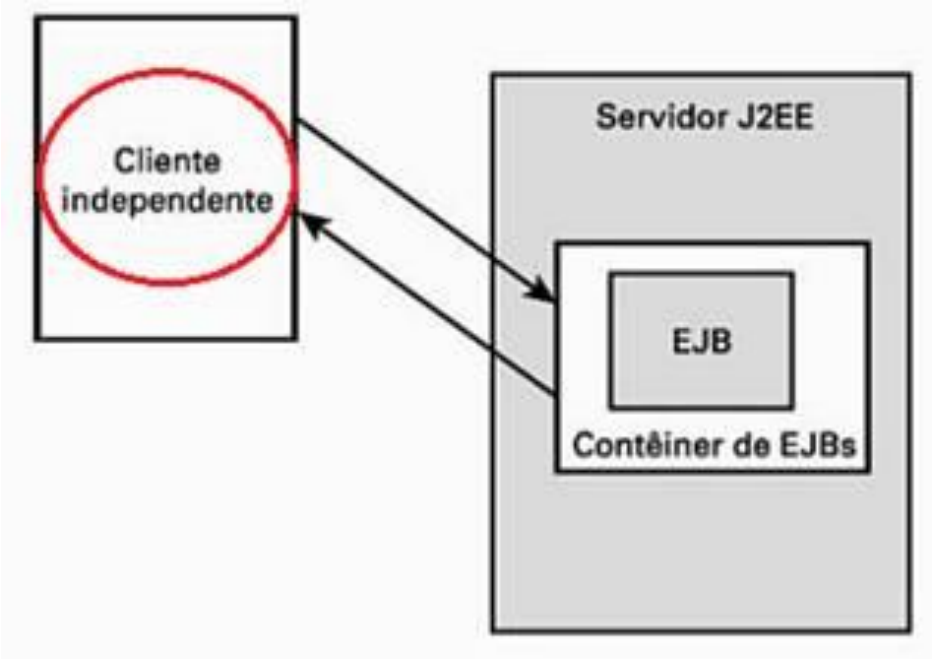
Camada Cliente

- **Cliente - Applet Java:** Os applets eram muito utilizados em sites de banco no início da internet, mas com o tempo caiu em desuso, pelo fato de que o cliente precisava ter uma cópia da **Java RunTime Enviroment** (JRE) instalado da mesma versão do site e além disso, possuir o plugin do applet Java no navegador para conseguir o acesso do banco.
  - O applet é um pequeno programa baseado em GUI que geralmente é executado dentro do navegador web, por isso que sempre é necessário que o navegador esteja com o plugin ativo para o funcionamento.
  - A invocação de um applet acontece no momento em que o cliente requisita uma página HTML que faz referência a um arquivo de classe (.class) no lado do servidor. Então, o servidor responde ao cliente retornando esse arquivo de classe e depois o applet é executado dentro da **JVM** fornecida pelo navegador.

Conhecendo os componentes no Java EE

Camada Cliente

- **Cliente Independente:** Esse tipo de cliente consegue acessar componentes através de seu contêiner. Por exemplo, na figura abaixo o cliente EJB consegue mandar os dados para um servlet que será gerada a resposta de acordo com a requisição feita pelo cliente.



Conhecendo os componentes no Java EE

Camada Cliente

- Em outros casos, esse tipo de cliente ignora a camada de apresentação e de negócio justamente para acessar diretamente os recursos do sistema de informações corporativas através do JDBC. No exemplo abaixo o cliente assume a responsabilidade pela camada da apresentação e de negócio.