

Lição 1



Introdução à Programação WEB



Objetivos

Ao final desta lição, o estudante será capaz de:

- Descrever como funciona a *WEB*
- Definir a arquitetura do tipo Cliente-Servidor
- Entender sobre o protocolo *HTTP*
- Definir o básico sobre a arquitetura *Java EE*
- Saber o que são *Servlets* e *Java Server Pages*



Porque migrar para a WEB?

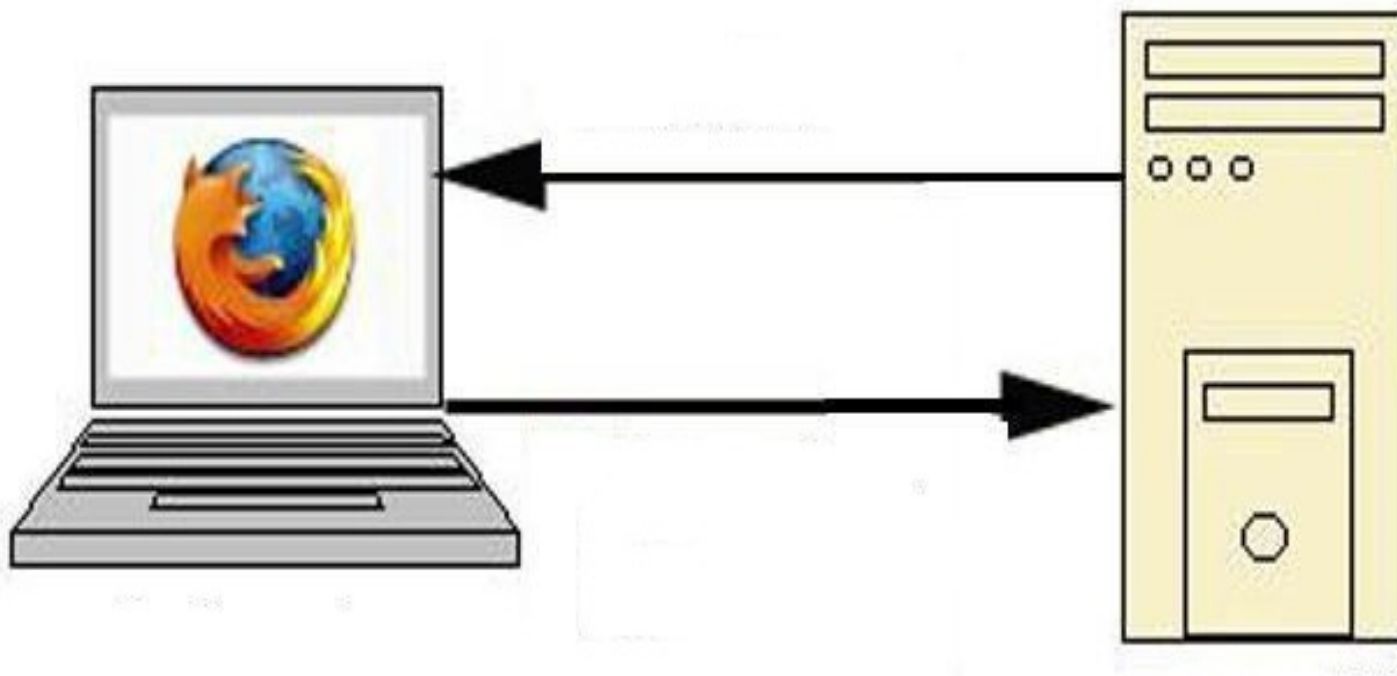
- Ambiente de tecnologia neutra
- Facilidade de distribuição e atualização



Arquitetura Cliente-Servidor

- Cliente pesado e cliente magro
- Arquitetura de cliente-servidor de uma perspectiva WEB
 - Aplicações de WEB usam clientes magros
 - Programa de cliente, um navegador neste caso, é só uma interface que o usuário usa para executar tarefas
 - Tudo mais reside no servidor

Arquitetura Cliente-Servidor



HTML

- Linguagem de marcação de hipertexto (***H**ypertext **M**arkup **L**anguage*)
- Conjunto de instruções para o navegador WEB que define como apresentar o conteúdo ao usuário
- Padrão aberto atualizado pelo *World Wide Web Consortium* (W3C)
- Navegadores conhecem HTML

HTTP

- Protocolo de Transferência de Hipertexto (***H**ypertext **T**ranfer **P**rotocol*)
- Um protocolo de rede com características específicas para WEB
- Roda sobre duas outras camadas de protocolo: TCP e IP

HTTP

- Usa uma seqüência de Requisição / Resposta
- Mensagens de requisição e resposta

Requisições HTTP

- Requisições do cliente para o servidor contêm:
 - Um cabeçalho inicial
 - Zero ou mais cabeçalhos adicionais
 - Um linha em branco
 - O corpo de mensagem (opcional)



Requisições HTTP: GET

- Método HTTP simples
- Usadas para pedir um recurso particular do servidor
- Usadas para enviar dados para o servidor



GET: URL Exemplo

`http://jedi-master.dev.java.net/servlets/NewsItemView?
newsItemID=2359&filter=true`

- Itens antes da marca de interrogação (?) é a URL original da requisição

`http://jedi-master.dev.java.net/servlets/NewsItemView`

- Parâmetros ou dados enviados para o servidor:
 - **Parâmetros** são codificados como pares de nome e valor
 - Formato: nome=valor
 - Mais de um conjunto de parâmetros, serão separados por um símbolo de &

`newsItemID=2359&filter=true`



Requisições HTTP: POST

- Projetado para que o navegador possa fazer:
 - Requisições complexas para o servidor
 - Requisições simples que requerem carregamento de arquivos para o servidor
- Esconde os dados dentro do corpo da mensagem que é enviada para o servidor

Resposta HTTP

- Contém o cabeçalho e o corpo de mensagem
- Conjunto diferente de cabeçalhos
- Tipo MIME-type

Páginas Dinâmicas ou Estáticas

- Conteúdo
 - Conteúdo estático não muda
 - Mudanças dinâmicas de conteúdo de acordo com a entrada do usuário
- Páginas dinâmicas têm mais flexibilidade e utilidade que estáticas

Páginas Dinâmicas ou Estáticas

- Alguns cenários onde conteúdo dinâmico é a única coisa que caberá a conta:
 - A página WEB está baseada nos dados submetidos pelo usuário
 - Os dados mudam freqüentemente
 - A página WEB usa informação de bancos de dados corporativo ou de outras fontes

Java Enterprise Edition (Java EE)

- Criada para o desenvolvimento de aplicações corporativas, baseada em componente
- Usa um Modelo de Aplicação de Multi-Camadas Distribuídas ou *multi-tier*

Java EE: Exemplo de aplicação multi-camada

- Uma aplicação WEB tem:
 - A camada de apresentação (navegador)
 - A camada lógica de negócio (aplicação que reside no servidor WEB)
 - A camada de armazenamento (o banco de dados com os dados da aplicação)
- As camadas de uma aplicação WEB estão distintamente separadas

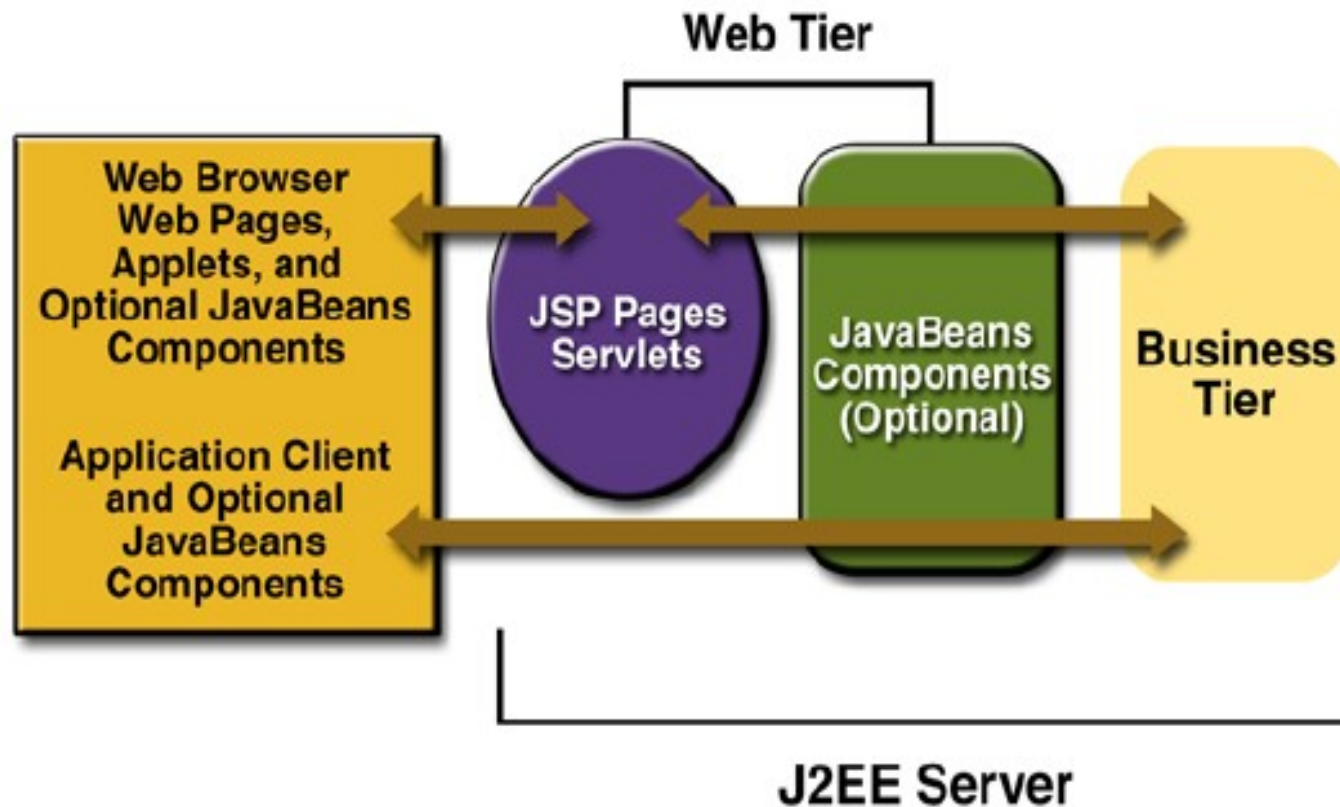


Java EE Visão Geral da Camada WEB

- Camada WEB
 - Uma das camadas na plataforma J2EE
 - Camada que interage com o navegador para criar conteúdo dinâmico
 - Tecnologias Java dentro desta camada:
 - Servlets
 - JavaServer Pages (JSP)



Java EE Visão Geral da Camada WEB



Camada WEB: Servlets

- Resposta primária do Java para adicionar funcionalidade aos servidores
- Ler dados contidos nas requisições passadas ao servidor e gerar uma resposta dinâmica
- Prover uma versão especial que implementa características específicas HTTP



Servlets Desvantagem

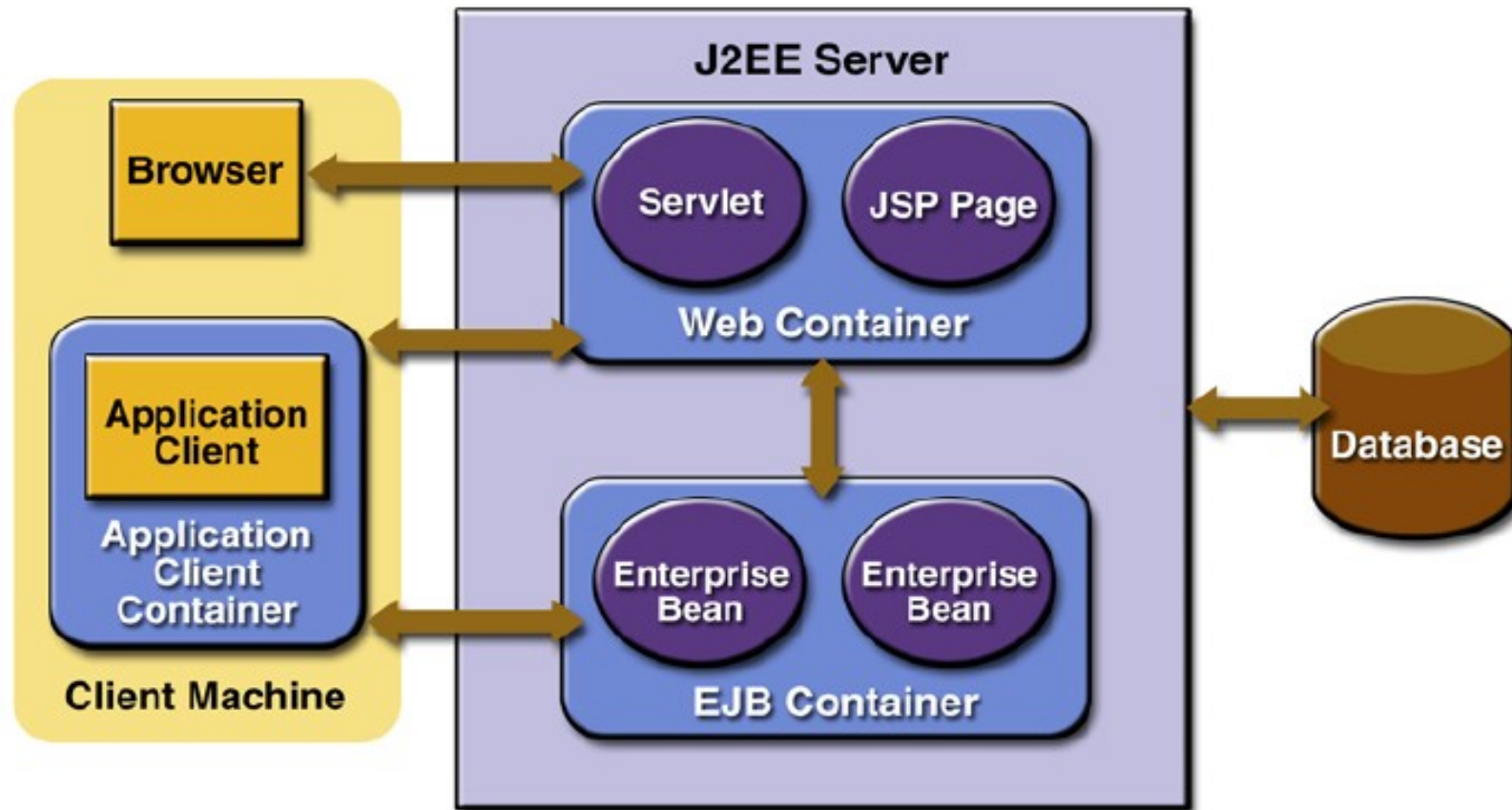
- *Servlets* são classes na linguagem Java
- Produzem saída imprimindo caracter como cadeias no fluxo de saída, neste caso a resposta HTTP



Camada WEB: JavaServer Pages

- Aspectos exatamente como HTML, só tem acesso a todas as capacidades dinâmicas de *servlets* com o uso de roteiros e linguagens de expressão
- Os projetistas podem concentrar-se em puro desenho HTML e simplesmente deixar marcadores para desenvolvedores preencherem com conteúdo dinâmico

Contêiner

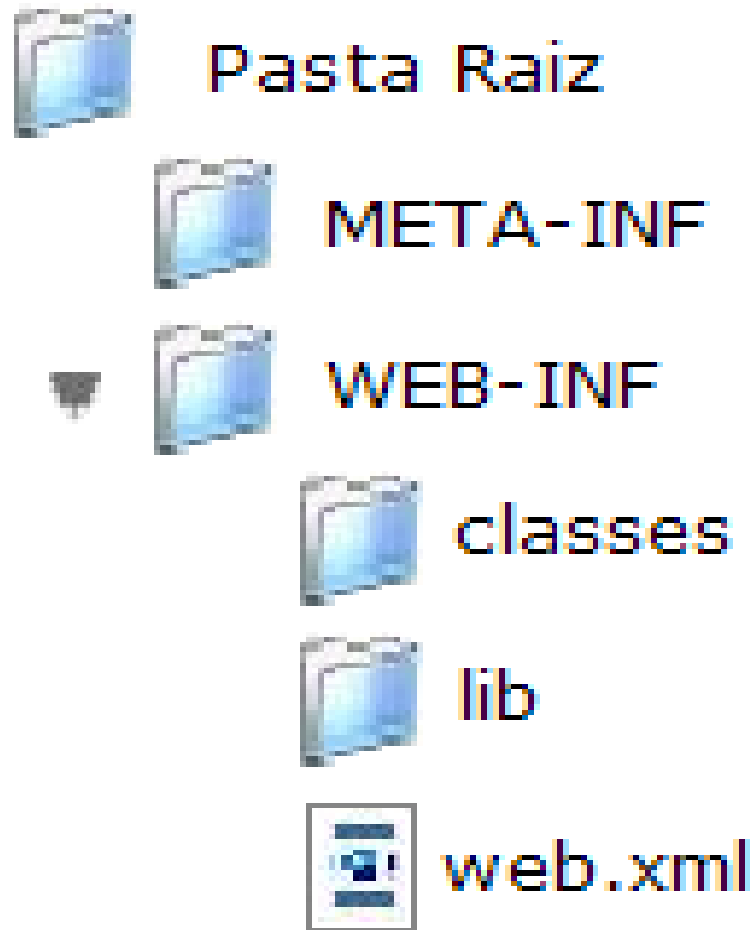


Contêiner

- Suporte de comunicações
- Gerência de ciclo de vida
- Suporte a multitarefa
- Segurança declarativa
- Suporte a JSP



Estrutura Básica de uma aplicação Java WEB



Estrutura Básica de uma aplicação Java WEB

- A pasta de alto nível NÃO precisa ser nomeada Raiz de Documento
- Qualquer outra pasta pode ser criada dentro desta estrutura de diretório
- Letras maiúsculas na pasta WEB-INF é intencional
- Todos os conteúdos da pasta WEB-INF não pode ser visto do navegador
- DEVE haver um arquivo web.xml dentro da pasta WEB-INF



Sumário

- Porque migrar para a WEB?
- Arquitetura Cliente-Servidor
- HTML e HTTP
- Páginas Dinâmicas ou Estáticas
- *Java Enterprise Edition*
- Estrutura básica de uma aplicação WEB



Parceiros

- Os seguintes parceiros tornaram JEDITM possível em Língua Portuguesa:



University of the Philippines
Java
Research and
Development
Center

