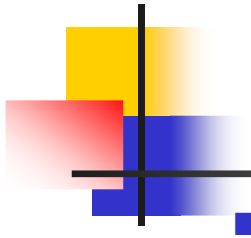
Java Enterprise Edition - JEE

03. Conceitos de J2EE

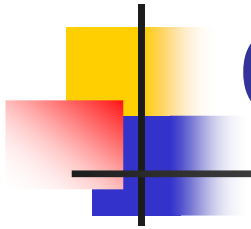
Esp. Márcio Palheta

gtalk: marcio.palheta@gmail.com



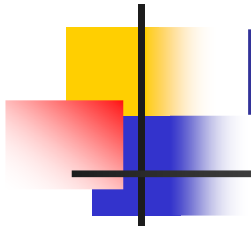
Agenda

- O que é o Java Enterprise Edition?
- APIs para J2EE
- Arquitetura de um WEB Site dinâmico
- Arquitetura J2EE
- Aplicações distribuídas
- Remote Method Invocation – RMI
- JAVA Naming Directory Interface – JNDI
- Aplicações em camadas
- O que é um container? E um servidor de aplicação?
- Qual a função do empacotamento
- Primeiro projeto WEB
- Análise do resultado final;
- Exercícios de fixação – Consultar



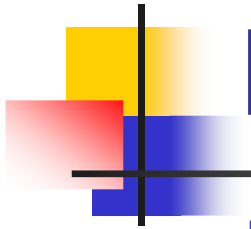
Conceitos fundamentais

- J2EE – Java 2 Enterprise Edition: é uma plataforma para desenvolvimento de aplicações distribuídas, baseadas em java;
- Você pode encontrar mais em: http://java.sun.com/javaee/
- J2EE ou JEE?: O nome J2EE foi utilizado até a versão 1.4 do java. Hoje, o termo correto é Java EE - JEE.



Principais APIs para JEE

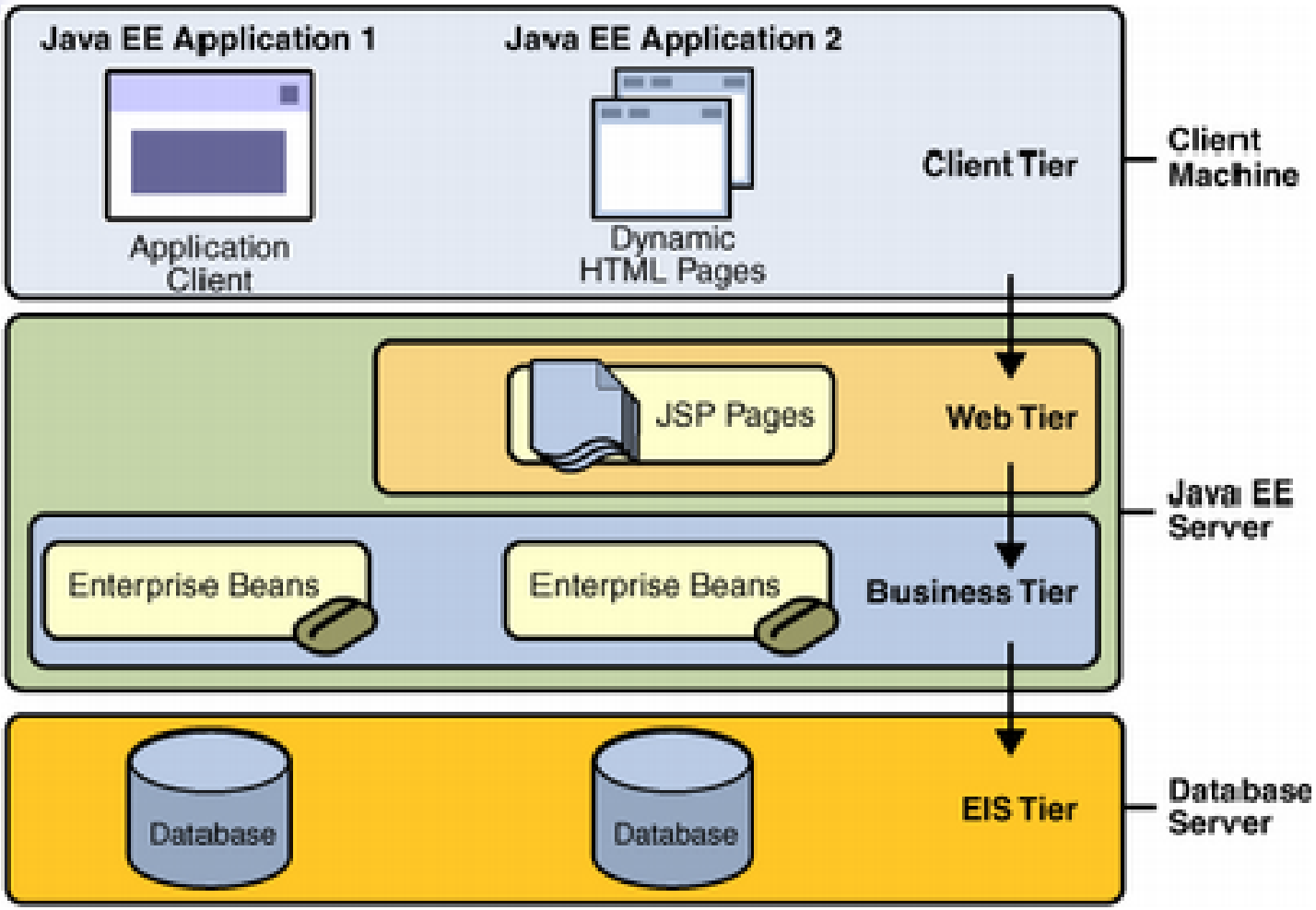
- JavaServer Pages (JSP), Servlets, Java Server Faces (JSF) (trabalhar para a web)
- Enterprise Javabeans (EJB) e Java Persistence Api(JPI) (objetos distribuídos, clusters, acesso remoto a objetos etc)
- Java API for XML Web Services (JAX-WS), Java API for XML Binding (JAX-B) (trabalhar com arquivos xml)
- Java Autenthication and Authorization Service (JAAS) (API padrão do Java para segurança)



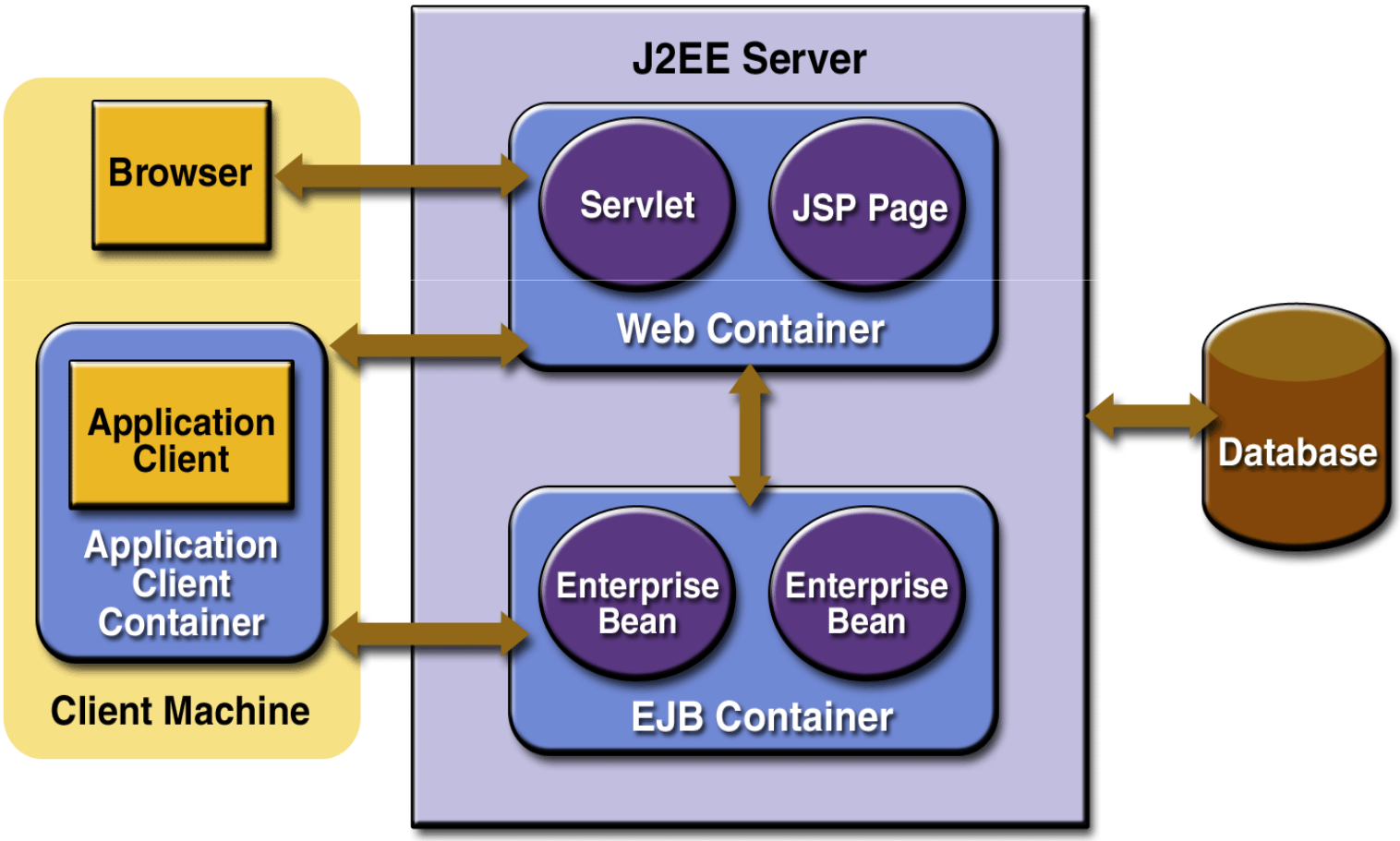
Principais APIs para JEE cont..

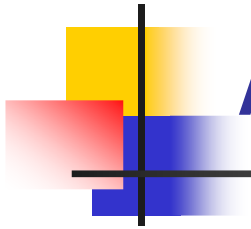
- Java Transaction API (JTA) (controle de transação no contêiner)
- Java Message Service (JMS) (troca de mensagens síncronas ou não)
- Java Naming and Directory Interface (JNDI) (espaço de nomes e objetos)
- Java Management Extensions (JMX) (administração da sua aplicação e estatísticas sobre a mesma)
- Entre outras para trabalhar com Webservices e outros tipos de acesso remoto ou invocação remota de métodos (RMI).

Arquitetura de um web site dinâmico



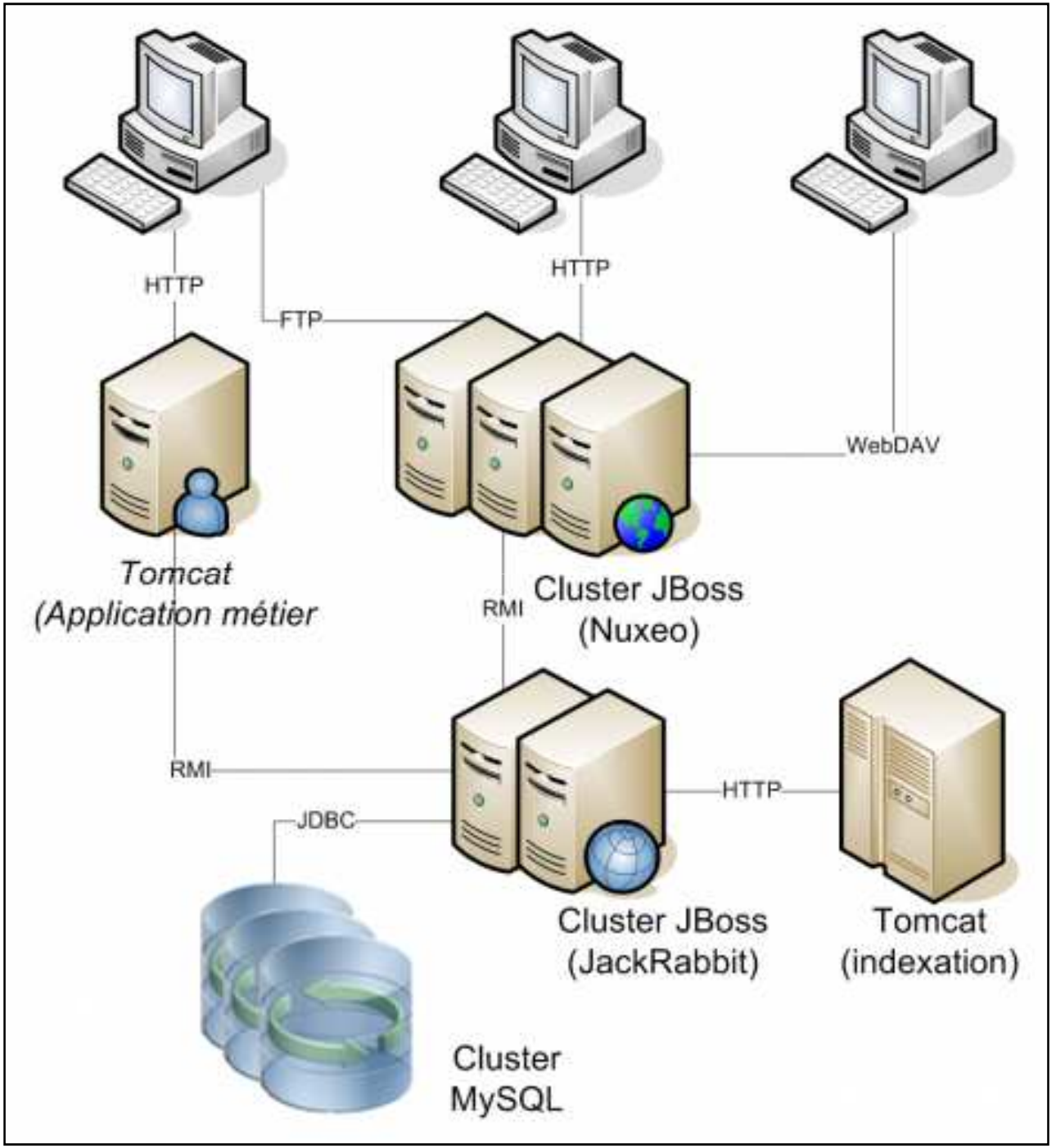
Arquitetura JEE



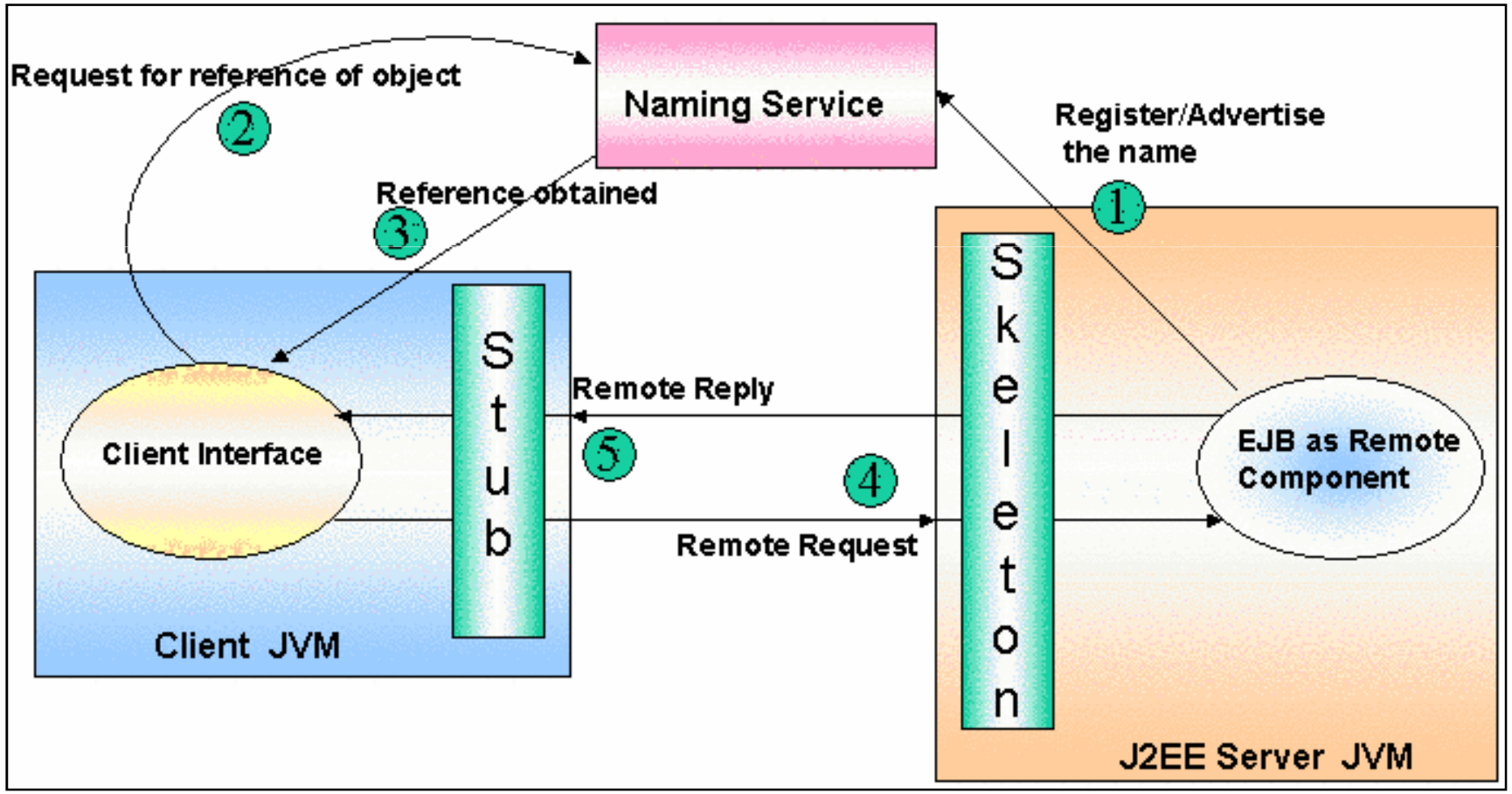


Aplicações distribuídas

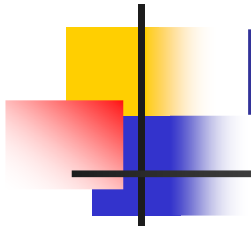
- Um aplicação que executa simultaneamente em várias máquinas;
- Um grupo de processos que executa em máquinas distintas e trabalha de forma coordenada e cooperativa para realizar uma determinada tarefa;
- Processos distintos, com atividades bem definidas;


HTTP
HTTP
FTP
WebDAV
Tomcat
(Application métier
RMI
Cluster JBoss
(Nuxeo)
RMI
HTTP
JDBC
Cluster JBoss
(JackRabbit)
Tomcat
(indexation)
Cluster
MySQL

Toda aplicação JEE implementa arquitetura distribuída

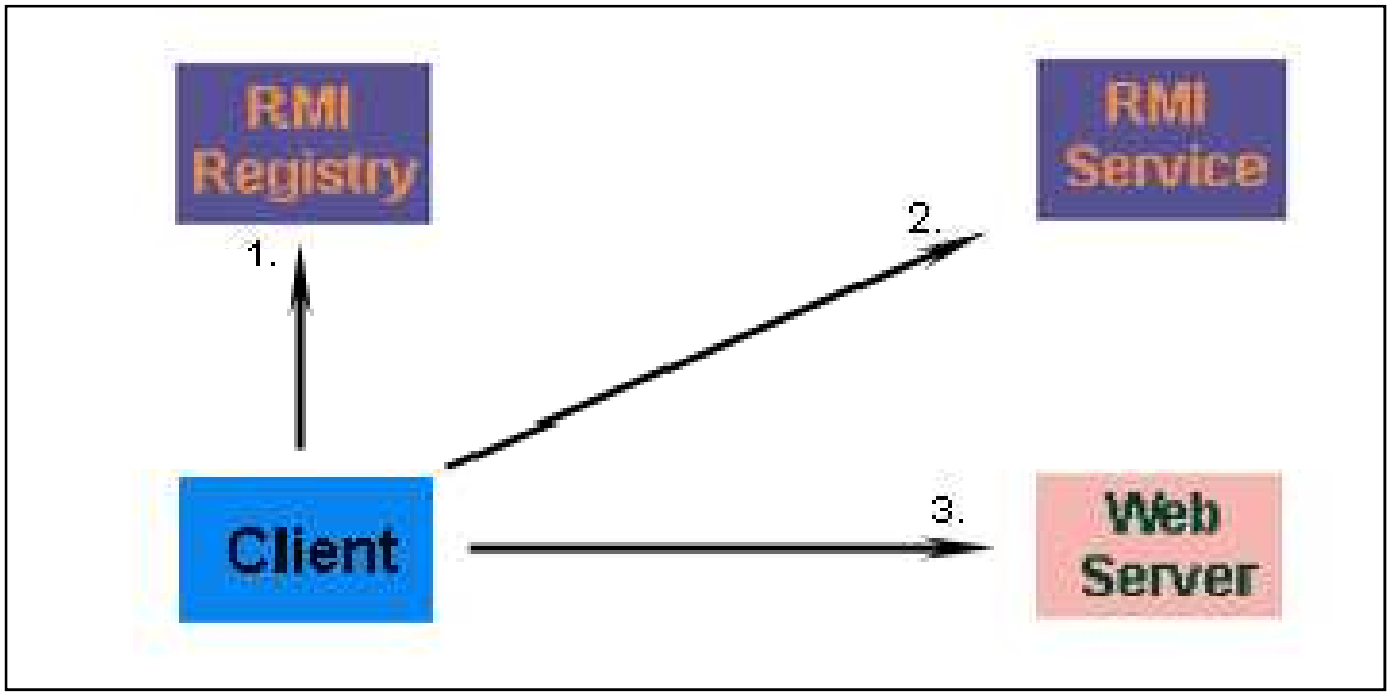


Arquitetura Java RMI – Remote Method Invocation

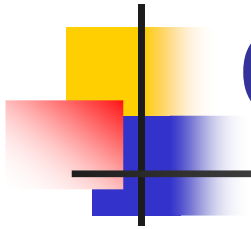


- Permite que aplicações JEE realizem chamadas a métodos de objetos remotos;
- Diferente de outros sistemas para execução remota, RMI permite que qualquer objeto JAVA seja utilizado;

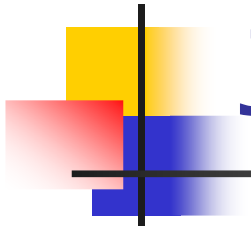
Arquitetura Java RMI – Remote Method Invocation



Java Naming Directory Components - JNDI

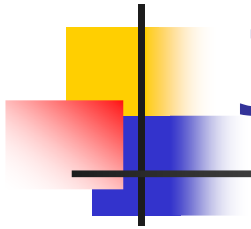


- O servidor JEE gerencia vários recursos e componentes que podem esta localizados tanto na própria máquina em que está rodando, quanto em outra máquina ou mesmo em outro servidor JEE.
- Para localizar um determinado recurso o JEE utiliza um servidor de nomes (Naming Service), que consiste de uma espécie de lista de telefônica onde os vários recursos são acessados através de um nome.



JNDI - continuação

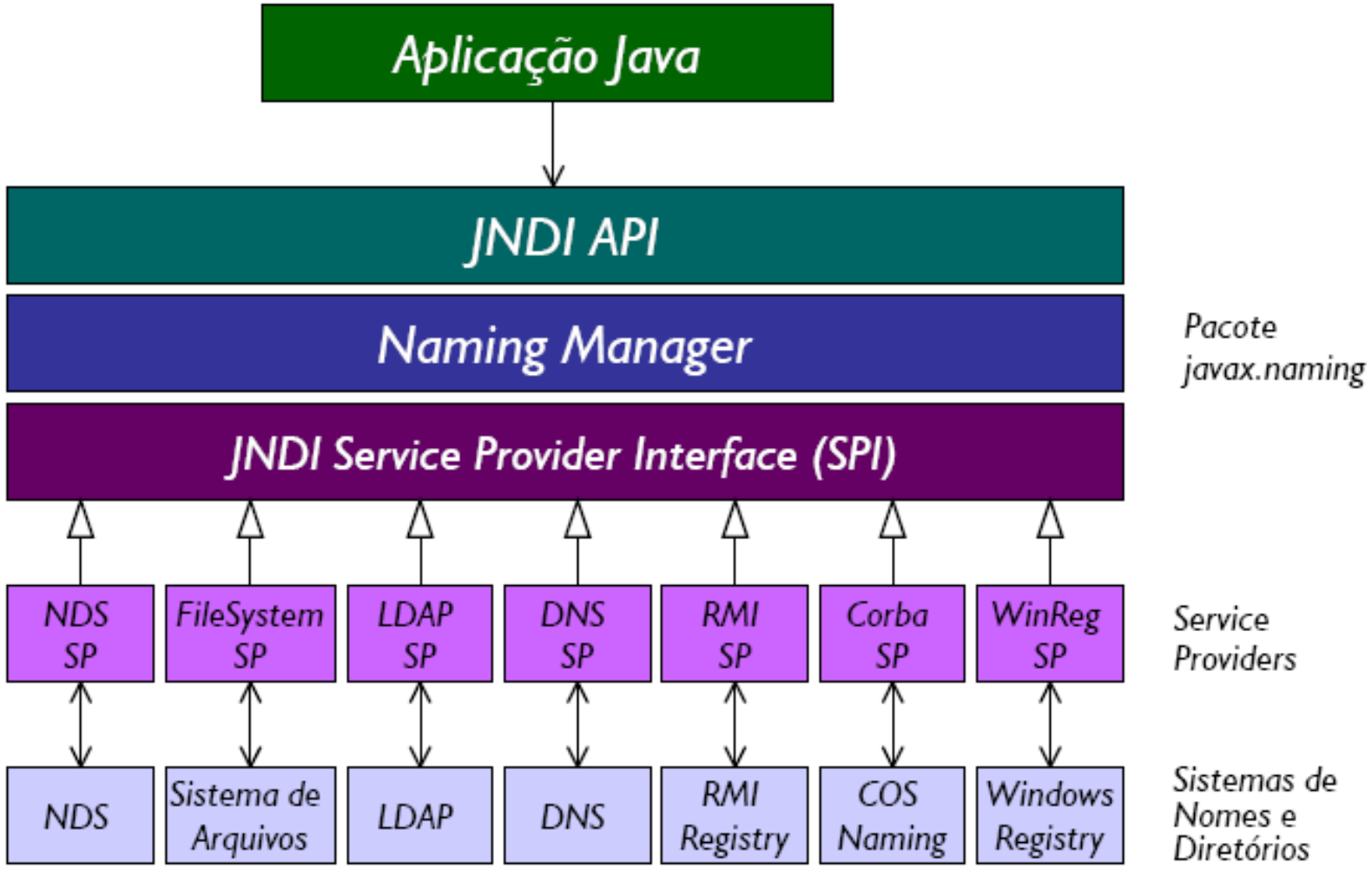
- O serviço que implementa esta lista é chamado de JNDI (Java Naming and Directory Interface).
- Toda vez que quisermos que um determinado recurso, como um banco de dados, um componente criado por nós ou por terceiros seja acessível para outros componentes ou aplicativos, devemos registrá-lo no JDNI.



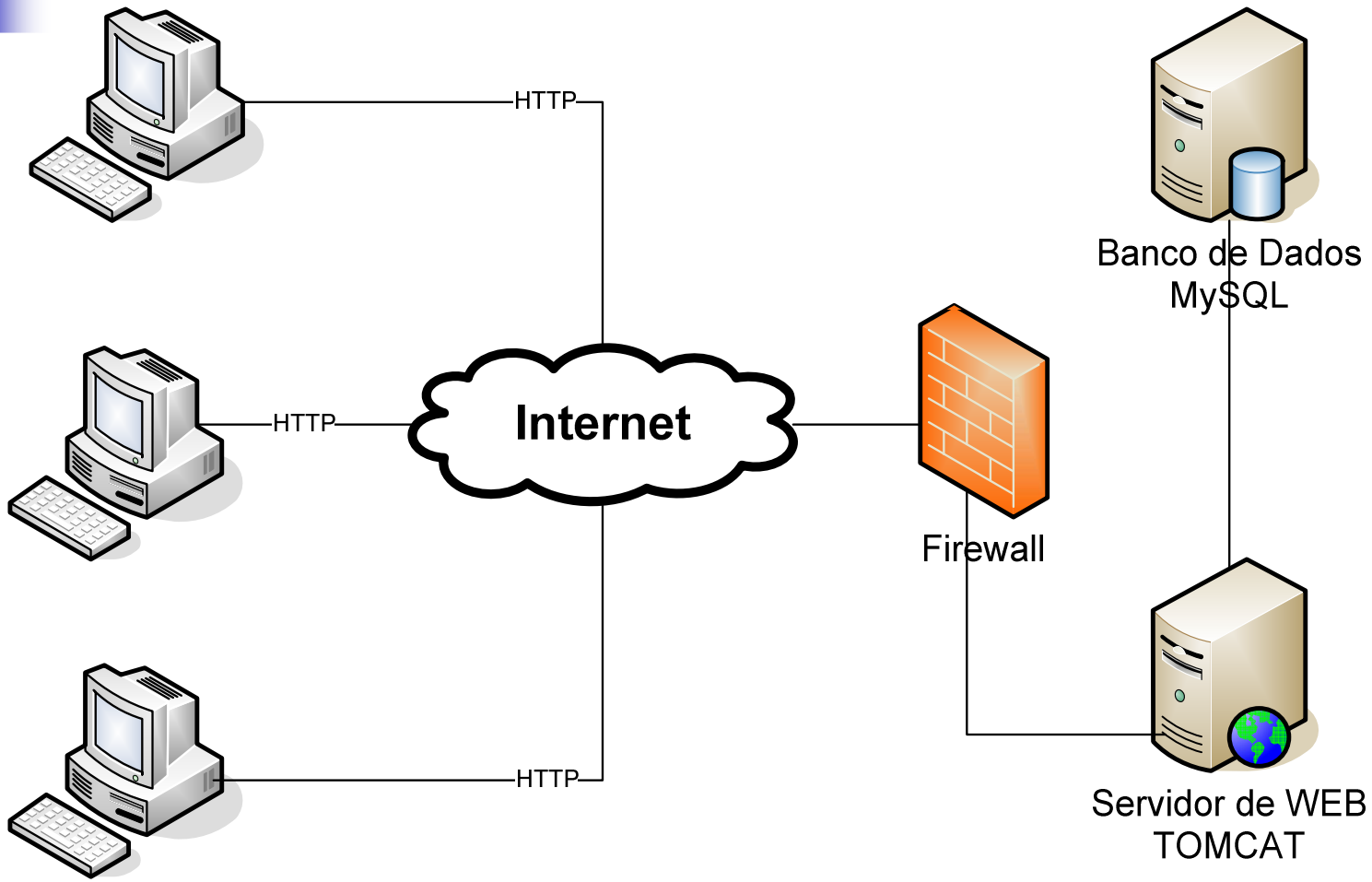
JNDI - Continuação

- Para que outros componentes utilizem um recurso compartilhado, devem obtê-lo a partir do JNDI, passando o nome com o qual foi registrado.
- O JNDI fornece um modo unificado de localização de recursos no JEE.

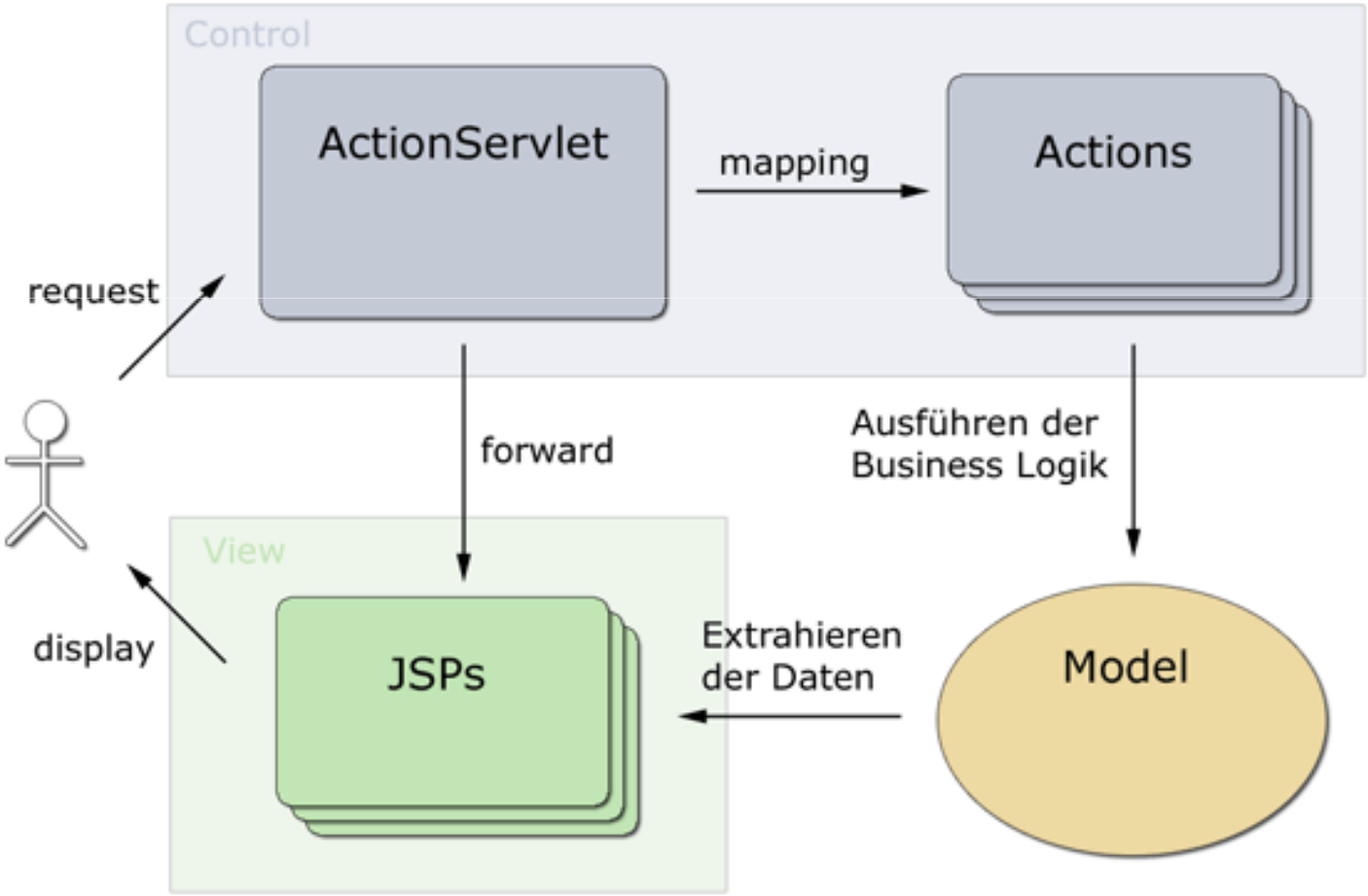
Java Naming Directory Components - JNDI

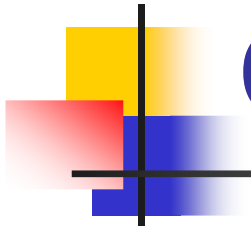


Infraestrutura WEB



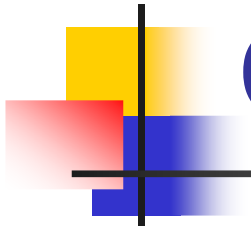
Desenvolvimento em camadas MVC – Model View Controller





Containers

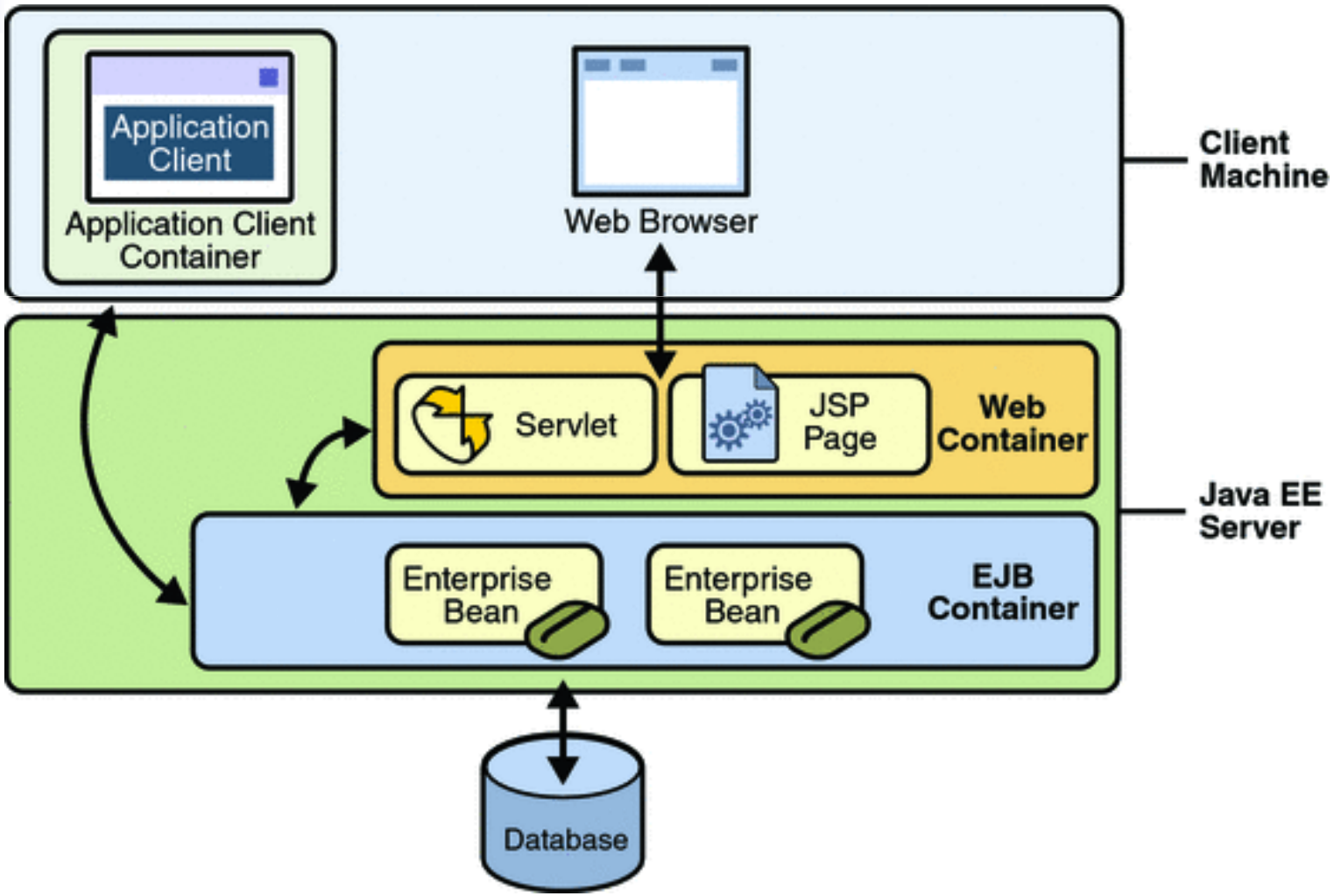
- Um container é a interface entre o componente e as camadas de baixo nível da plataforma onde roda.
- É uma espécie de Sistema Operacional para objetos.
- Antes que um componente EJB ou Web possa ser executado em um container JEE, ele precisa ser implantado (deployed) no container;

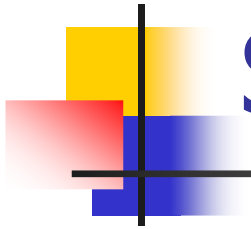


Containers - continuação

- O container é responsável por chamar os métodos (callback) que controlam o ciclo de vida dos componentes;
- O container também é quem serve de interface para que o componente utilize serviços de middleware implícito, declarados nos seus arquivos de configuração;
- A plataforma JEE define dois tipos principais de containers:
  - Container Web
  - Container EJB

Containers WEB e EJB

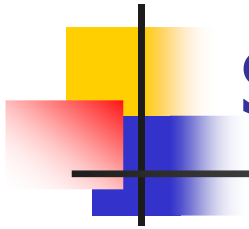




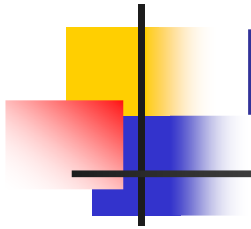
Servidores de aplicação

- Oferecem ambientes para a operação de componentes implantados em containers;
- Oferecem diversos serviços de middleware, como:
  - Autenticação e autorização;
  - Gerência de recursos;
  - Persistência;
- Ex: JBoss, Sun JEE Server e OAS

Serviços oferecidos por um servidor de aplicações



- Java Database Connectivity (JDBC) que oferece acceso a sistemas de banco de dados
- Java Transaction API (JTA) ou Java Transaction Service (JTS) que proporciona suporte para transacões a componentes JEE.
- Java Messaging Service (JMS) para comunicação assíncrona entre componentes JEE.
- Java Naming e Directory Interface (JNDI) que proporcionam acesso a nomes e diretórios.

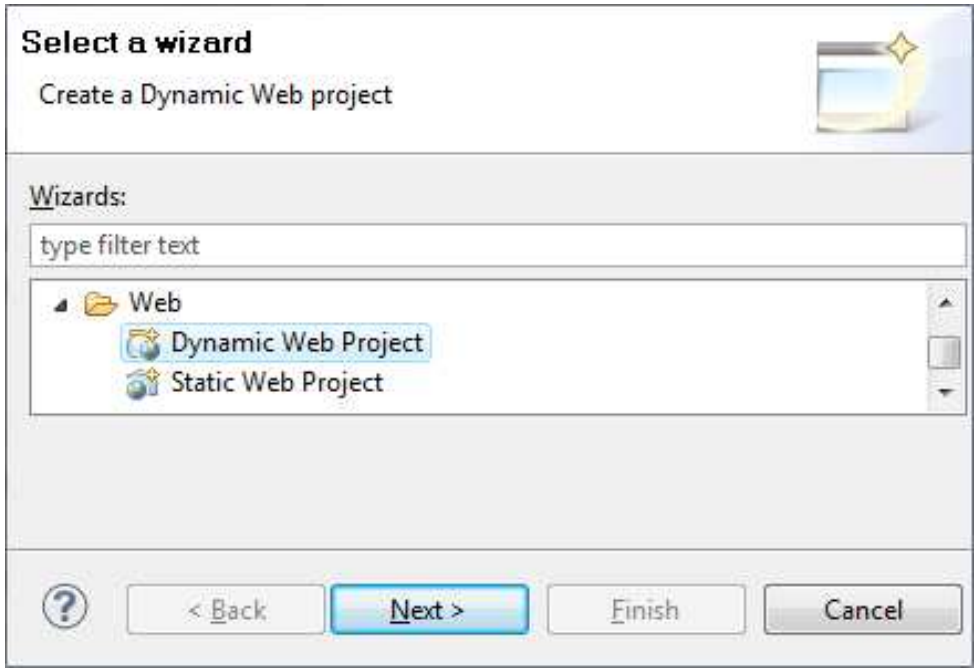


Empacotamento de aplicações

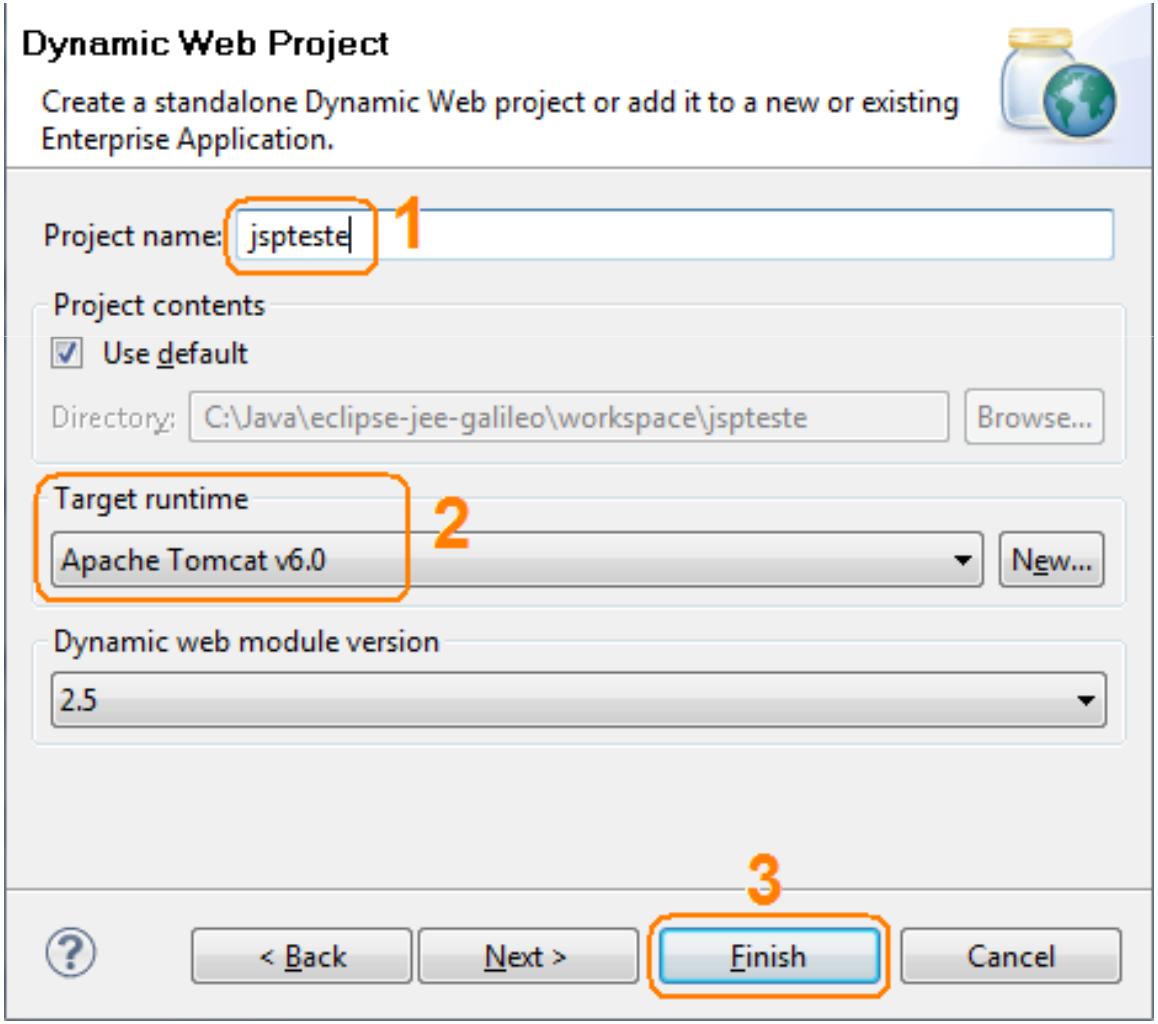
- Para que uma aplicação JEE fique disponível ao acesso de clientes, é necessária sua publicação em um servidor JEE.
- Enterprise Archive (.ear);
- Java Archive (.jar);
- Web Archives (.war)

Primeiro projeto WEB

- Clique em: New / Project / Web
- Selecione: Dynamic Web Project e clique em Next>;

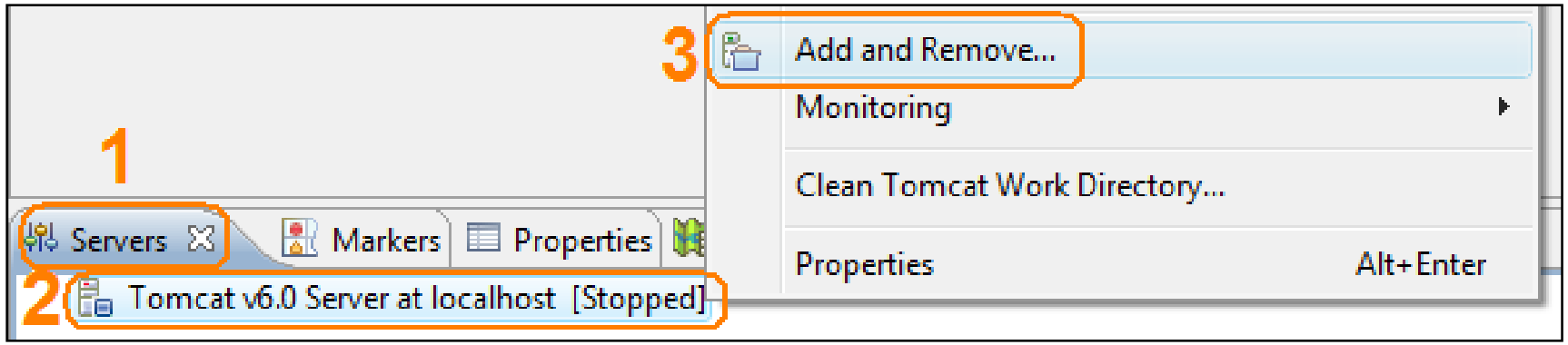


Aplicação jspteste



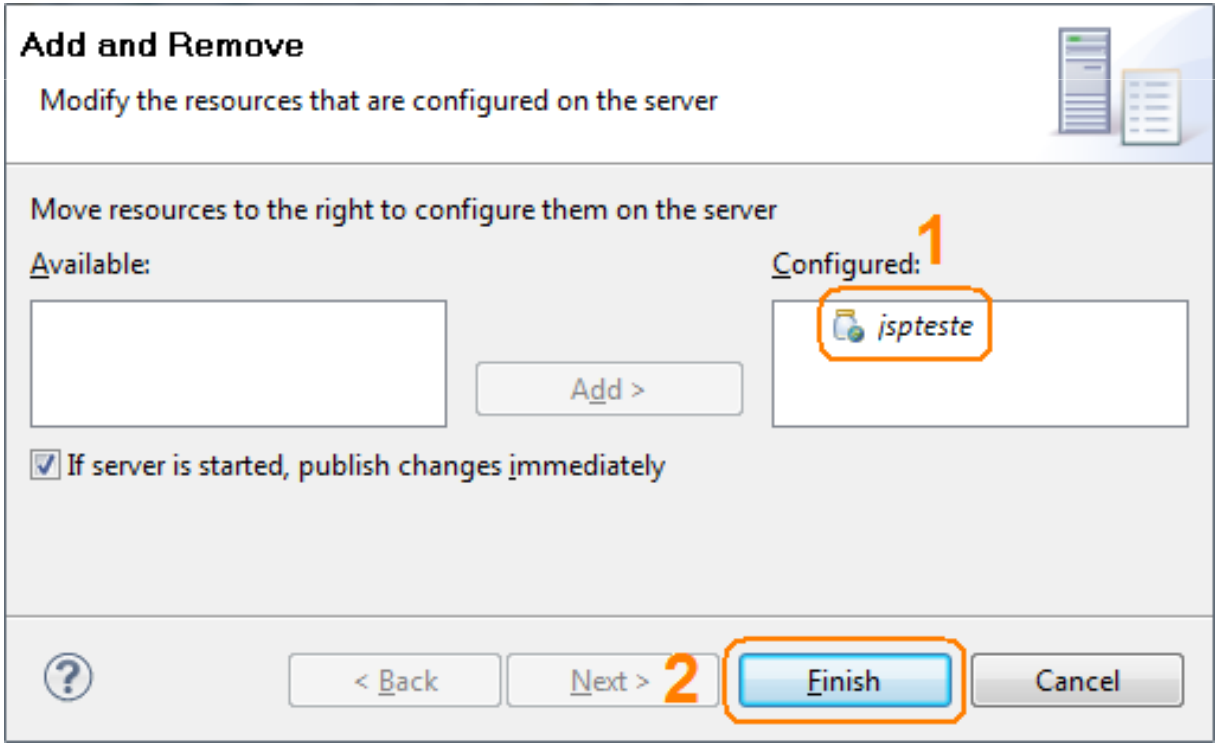
Configuração do Servidor WEB

- Na aba Servers, clique com o botão direito no Tomcat e selecione a opção Add and Remove Projects:



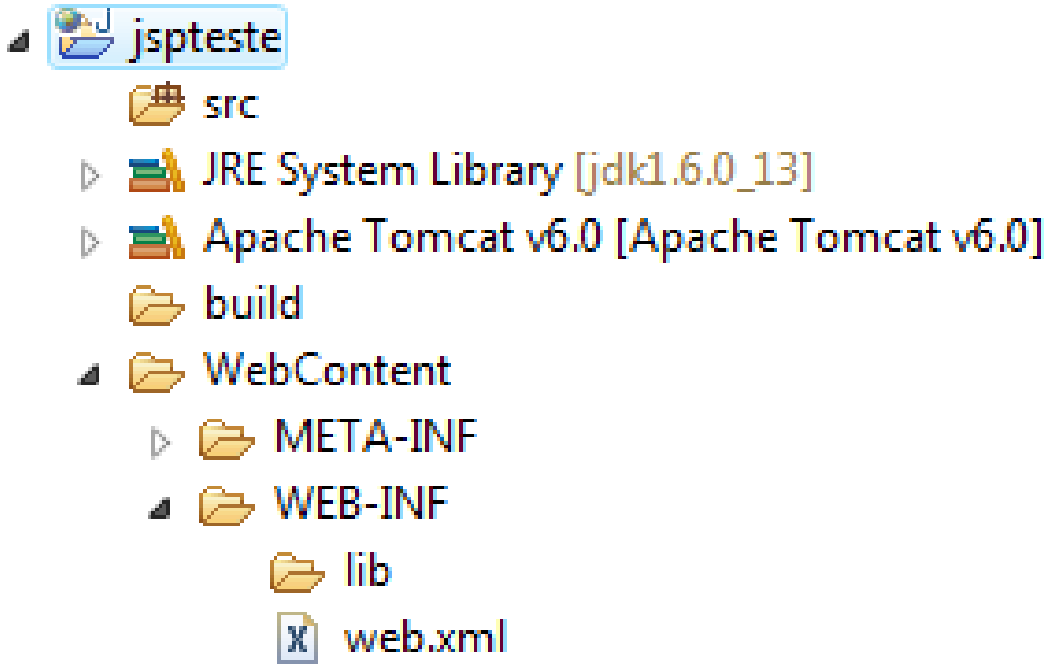
Configuração do servidor WEB

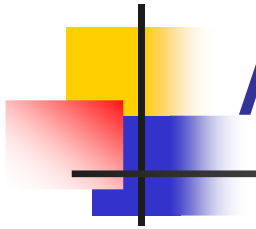
- Adicione o projeto jspteste à lista de projetos configurados:



O que foi gerado?

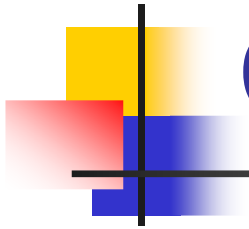
- O eclipse gera a estrutura padrão de organização de pacotes em um projeto WEB





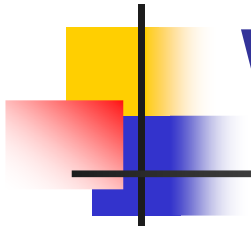
Análise da estrutura

- src: Pasta onde ficarão os códigos java
- build(bin): pasta onde são armazenados arquivos .class;
- Em JEE, as aplicações estão organizadas em ambientes distintos, conhecidos como contextos web;
- Por padrão, o eclipse gera o context name com o mesmo nome do projeto: jspteste



Configurações WEB

- Para acessarmos a aplicação, podemos utilizar a url:
  - http://localhost:8080/jspteste
- WebContent: é a pasta padrão para armazenamento de arquivos que podem ser acessados a partir da URL da aplicação;
- Ex.:Um arquivo hello.html é acessado: http://localhost:8080/jspteste/hello.html

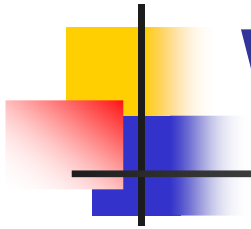


WEB-INF

- Pasta extremamente importante para qualquer projeto web JEE;
- web.xml: Contem configurações e recursos necessários à execução do projeto no servidor;
- A seguir, será apresentada a estrutura inicial do arquivo web.xml;

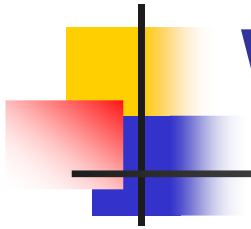
web.xml – estrutura inicial

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns="http://java.sun.com/xml/ns/javaee"
    xmlns:web="http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd"
    xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
                http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd"
  id="WebApp_ID" version="2.5">
  <display-name>jspteste</display-name>
  <welcome-file-list>
    <welcome-file>index.html</welcome-file>
    <welcome-file>index.htm</welcome-file>
    <welcome-file>index.jsp</welcome-file>
    <welcome-file>default.html</welcome-file>
    <welcome-file>default.htm</welcome-file>
    <welcome-file>default.jsp</welcome-file>
  </welcome-file-list>
</web-app>
```



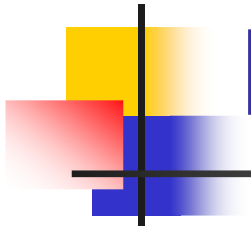
WEB-INF/lib

- O diretório lib dentro do WEB-INF pode conter todas as bibliotecas necessárias para a aplicação web, evitando, assim, que o classpath da máquina que roda a aplicação precise ser alterado.
- Cada aplicação web poderá usar suas próprias bibliotecas, com suas versões específicas!
- Evite o uso do classpath global.



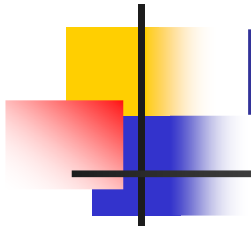
WEB-INF/classes

- Para rodarmos nossa aplicação no servidor, precisamos ter acessado às classes compiladas.
- O eclipse compila nossas classes na pasta **build** e depois, automaticamente, copia as coisas para o **WEB-INF/classes**.
- Imagine se o usuário tiver acesso a essa pasta! Códigos compilados (facilmente descompiláveis), bibliotecas potencialmente sigilosas, arquivos de configuração internos etc.
- Para que isso não aconteça, a pasta **WEB-INF** com esse nome especial é uma **pasta invisível ao usuário final**.
- Tente acessar a URL http://localhost:8080/jspteste/WEB-INF



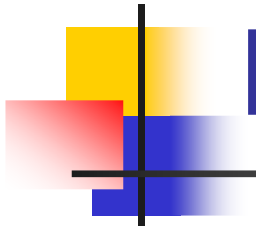
Resumo da estrutura

- src - código fonte Java (.java)
- build – onde ocorre a compilação (.class)
- WebContent - content directory (páginas, imagens, css etc vão aqui)
- WebContent/WEB-INF/ - pasta oculta com configurações e recursos do projeto
- WebContent/WEB-INF/lib/ - bibliotecas .jar
- WebContent/WEB-INF/classes/ - arquivos compilados são copiados para cá



Exercício 01

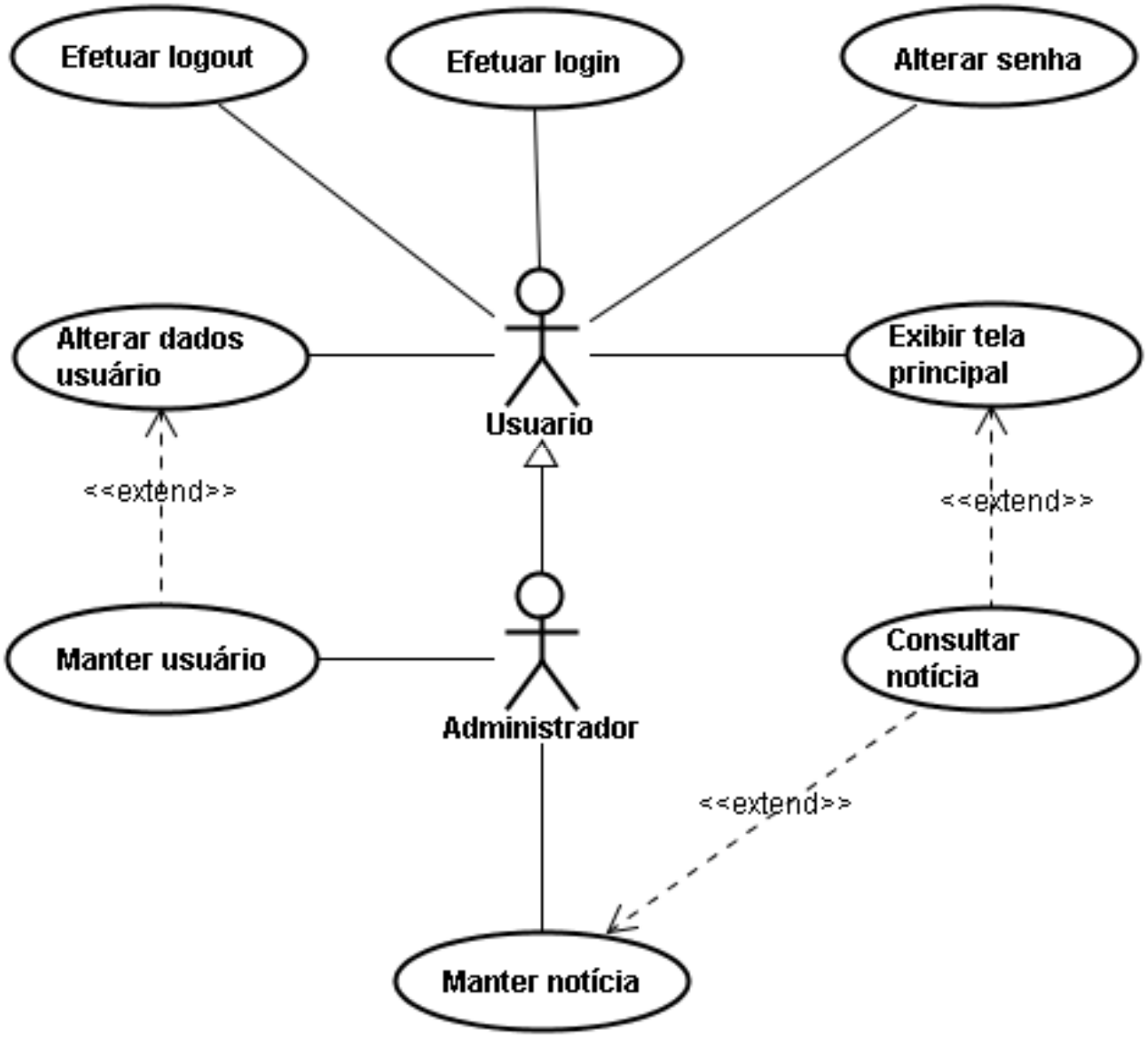
- Crie o arquivo **WebContent/index.html** com o seguinte conteúdo:
  - <html>
    - <h1>Novo projeto jspteste</h1>
  - </html>
- 2) Inicie (ou reinicie) o Tomcat clicando no botão de play verde na aba Servers.
- 3) Acesse no navegador: http://localhost:8080/jspteste/index.html
- Teste também a configuração do welcome-file: http://localhost:8080/jspteste/



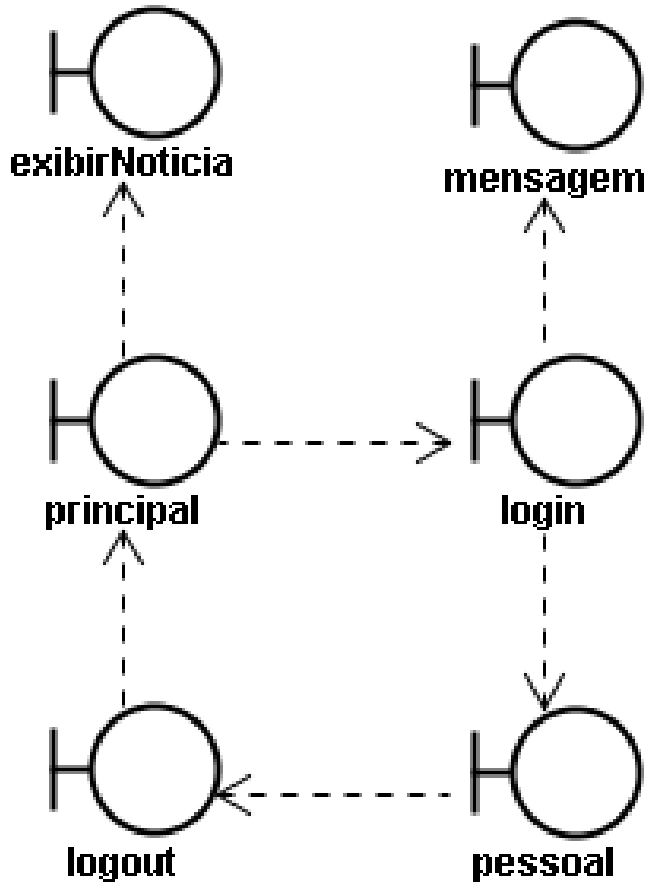
Exercício 02 – Estudo de caso

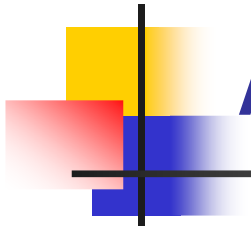
- A empresa MP News atua no mercado local de divulgação de notícias em mídia impressa;
- A empresa deseja criar um site onde possa publicar suas notícias, a fim de diminuir os gastos com impressões;
- Você foi contratado para implementar o site, a partir de artefatos gerados por analistas de sistemas da empresa

Diagrama de Casos de Uso



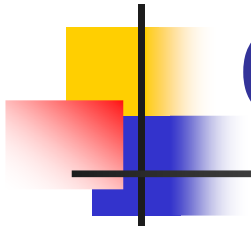
MP News – Diagrama navegacional





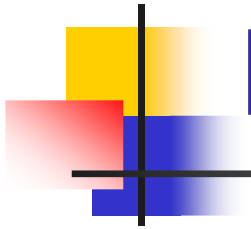
Atividades do projeto final

- Definição do tema;
- Levantamento de requisitos
- Entrevista com cliente;
- Delimitação do escopo;
- Diagrama de contexto;
- Especificação da visão geral do projeto



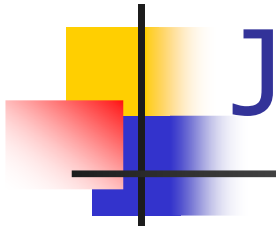
O que vem a seguir?

- Persistência em java
- Acesso a banco de dados
- Java DataBase Connectivity – JDBC;
- Padrões de projeto;
- Factory;
- DAO;
- Servlets;



Referências

- www.caelum.com.br
- Hall, Marty, “Core Servlets and Java Server Pages”, Janeiro 2002, Sun Microsystems Press;
- http://java.sun.com/j2ee/1.6/docs/tutorial/doc/index.html
- http://java.sun.com/products/jndi/docs.html
- http://java.sun.com/blueprints/corej2eepatterns/Patterns/index.html



Java 2 Enterprise Edition - J2EE

03. Conceitos de J2EE

Esp. Márcio Palheta

gtalk: marcio.palheta@gmail.com