

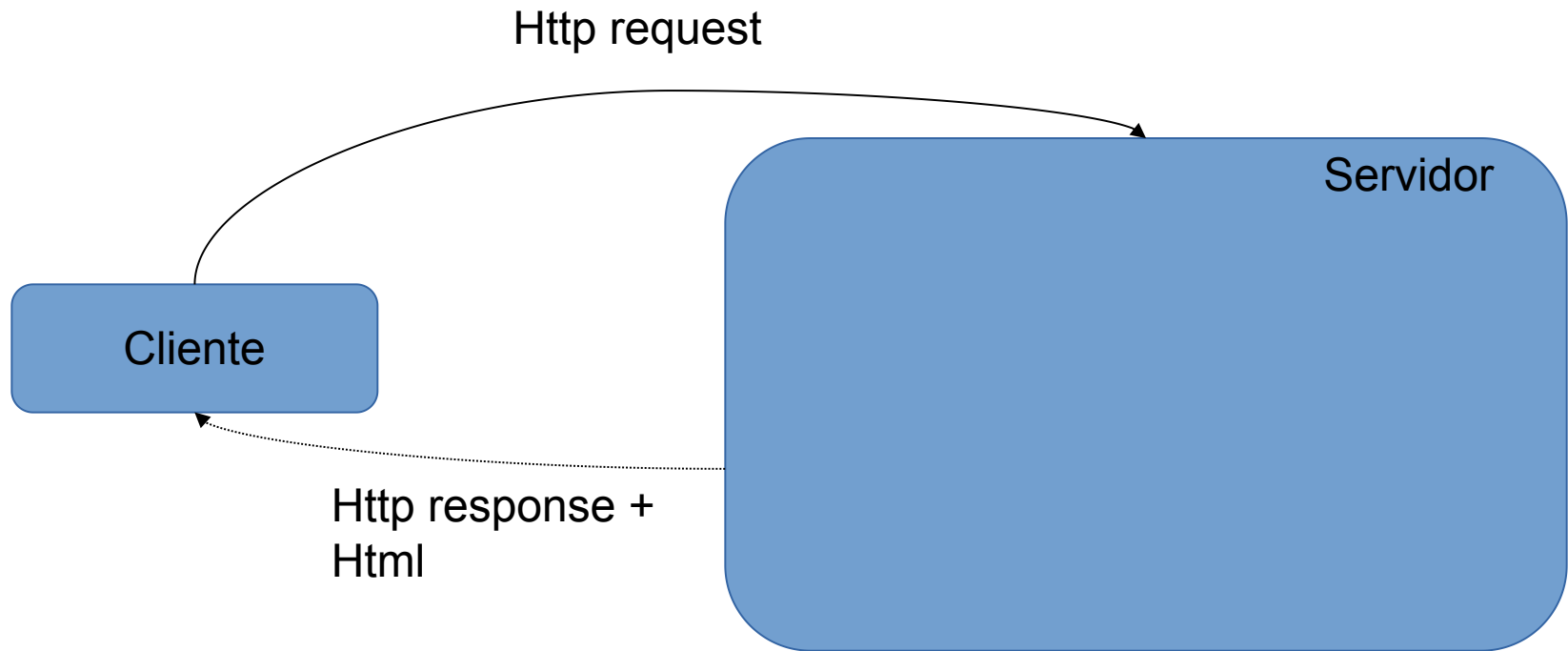
Arquitetura da Aplicação Web e J2EE



Desenvolvimento Web

Arquitetura da Aplicação Web

Visão Geral

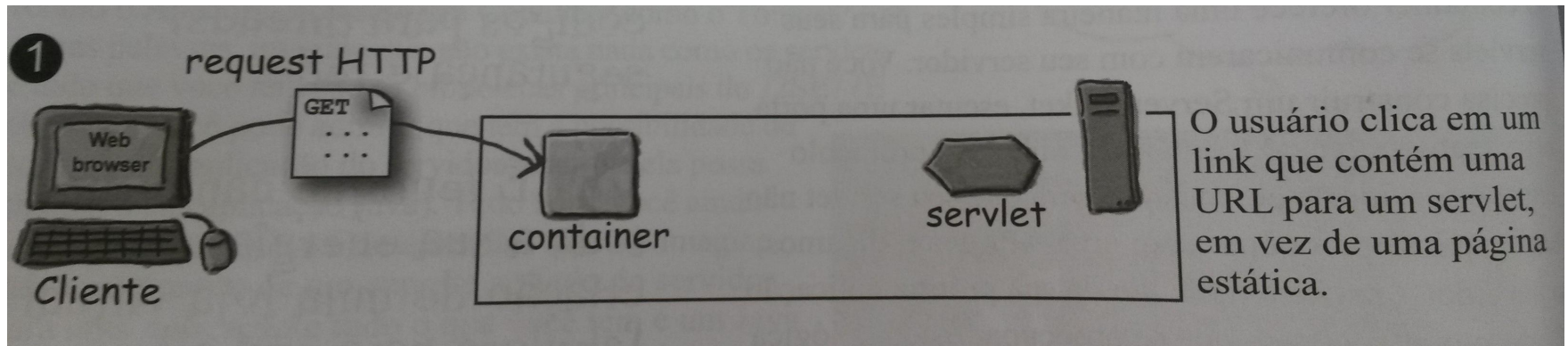


Container

- ↘ Suporte para comunicações
 - Abstrai os aspectos de comunicação com o servidor (comunicação com portas, sockets, etc).
- ↘ Gerenciamento do ciclo de vida
 - Inicialização, morte, passagem de parâmetros via http etc.
- ↘ Suporte a multithread
 - Controla a instanciação e acessos múltiplos às threads associadas aos servlerts
- ↘ Segurança
 - Tratamento no container (refletido nas aplicações embarcadas no container)
- ↘ Suporte a JSP (Java)

Container: Como Funciona

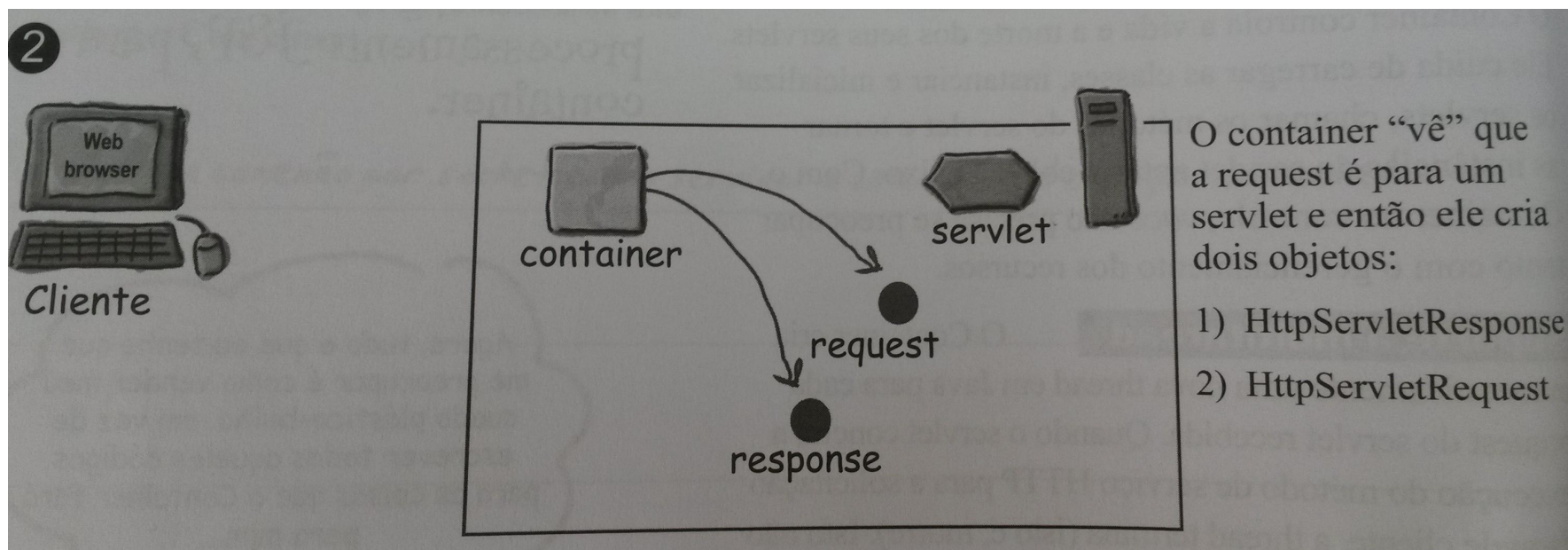
Passo 1



(BASHAM et al., 2011)

Container: Como Funciona

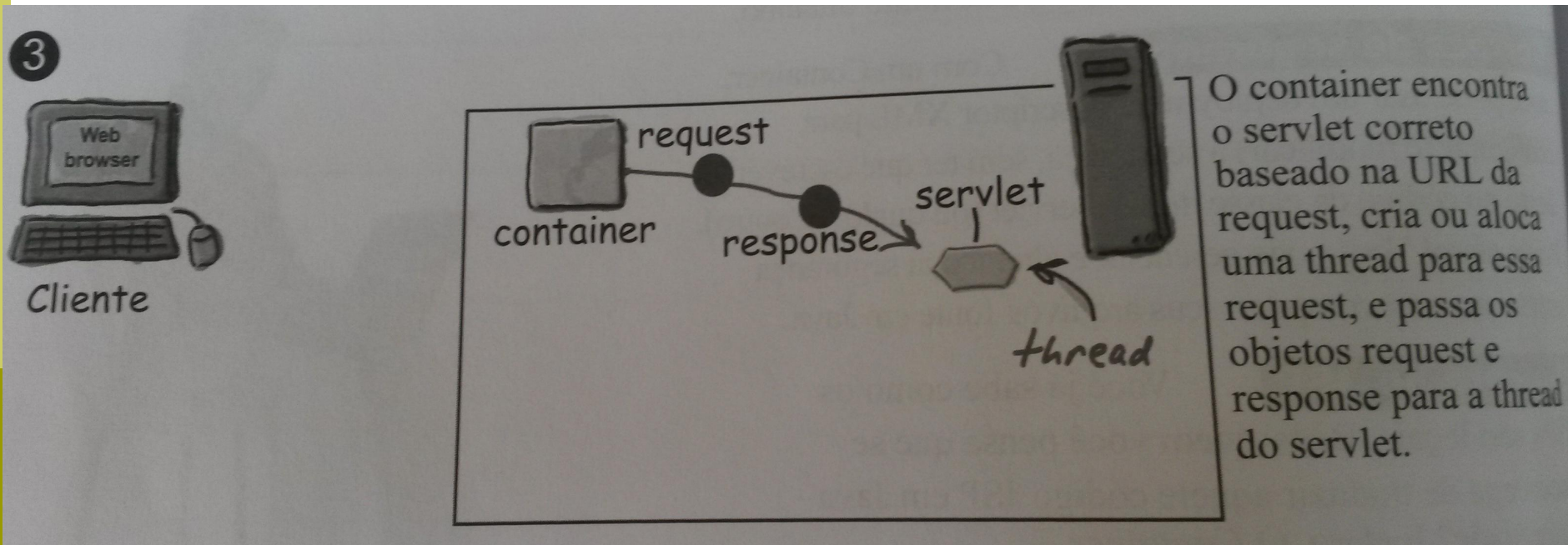
Passo 2



(BASHAM et al., 2011)

Container: Como Funciona

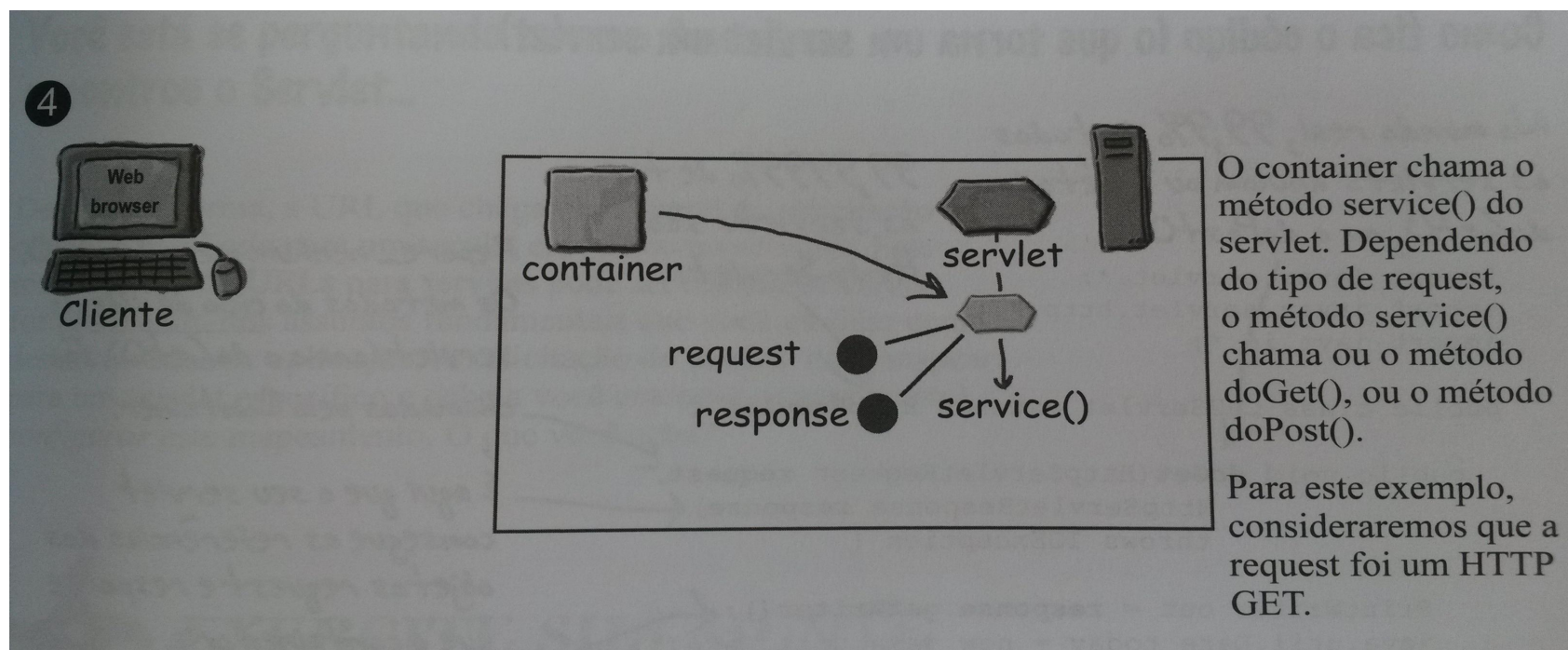
Passo 3



(BASHAM et al., 2011)

Container: Como Funciona

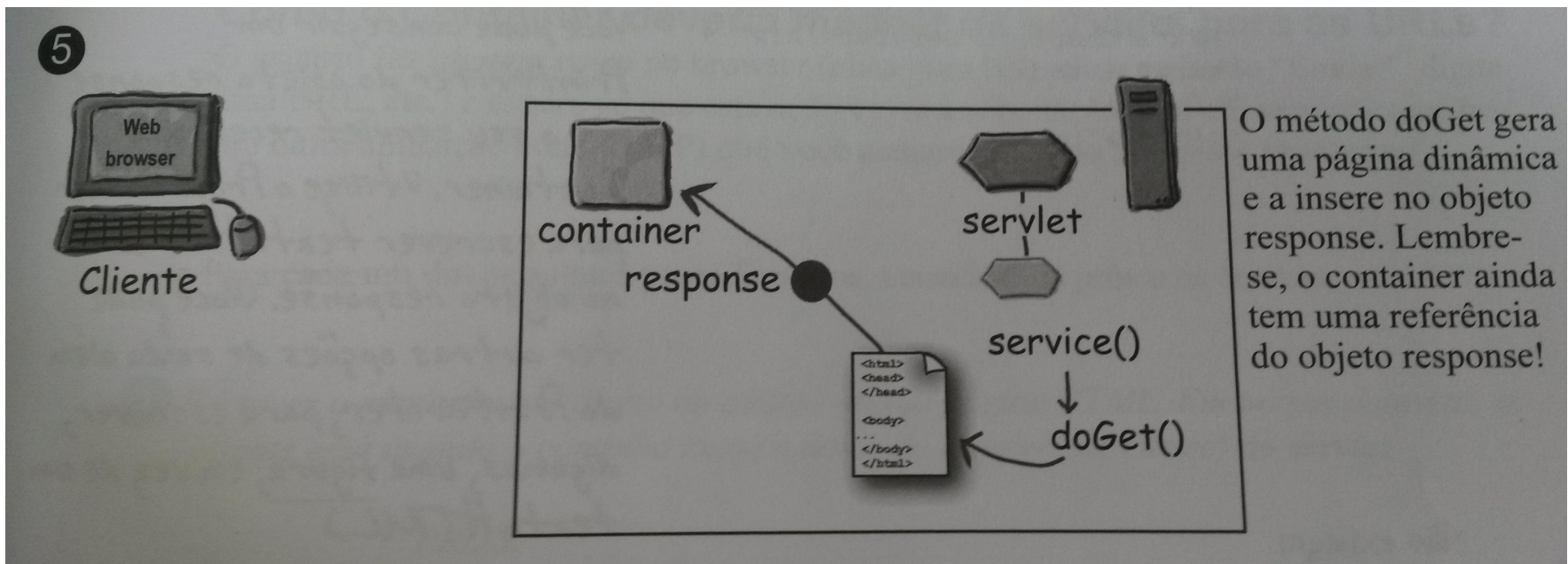
Passo 4



(BASHAM et al., 2011)

Container: Como Funciona

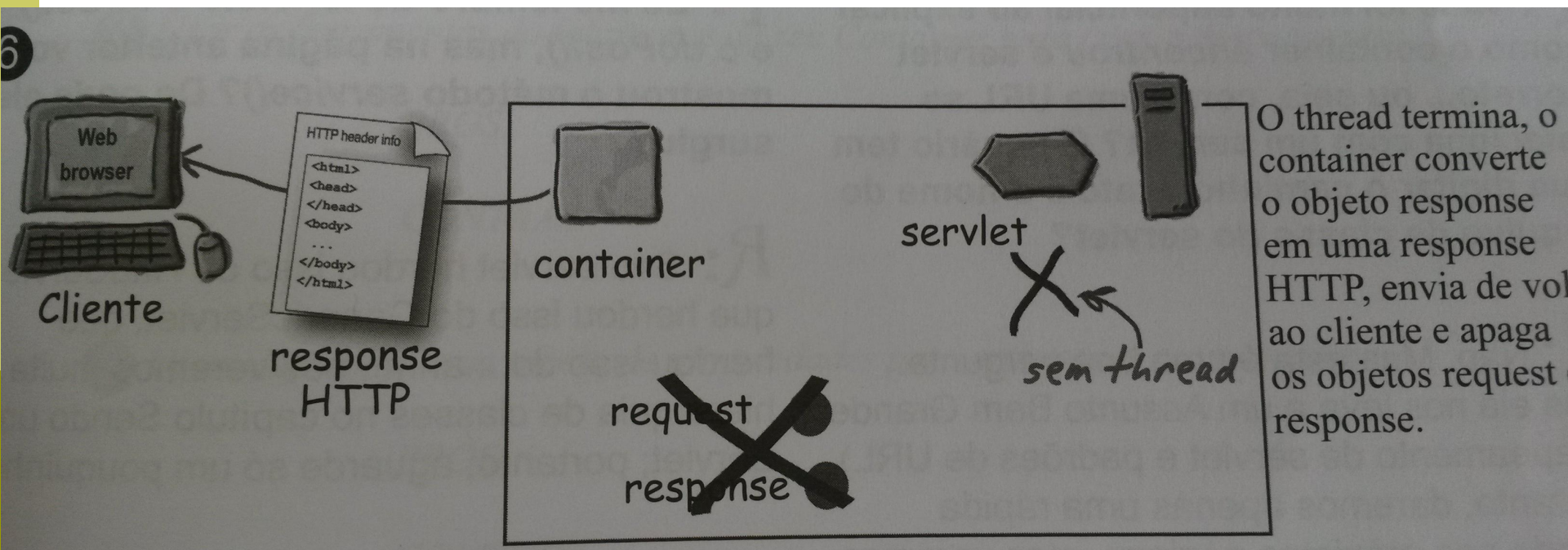
Passo 5



(BASHAM et al., 2011)

Container: Como Funciona

Passo 6



(BASHAM et al., 2011)

J2EE

(Java Enterprise Edition)

Introdução

- ↵ Conjunto de especificações indicando como uma determinada parte da plataforma Java deve ser implementada
- ↵ Qualquer um pode implementar uma especificação Java (Inclusive você!)
- ↵ Independência de fabricante

Algumas Especificações

- ↘ JavaServer Pages (JSP), Java Servlets, Java Server Faces (JSF) (trabalhar para a Web, onde é focado este curso)
- ↘ Enterprise Javabeans Components (EJB) e Java Persistence API (JPA). (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 e webservices)
- ↘ Java Authentication and Authorization Service (JAAS) (API padrão do Java para segurança)
- ↘ Java Transaction API (JTA) (controle de transação no contêiner)
- ↘ Java Message Service (JMS) (troca de mensagens assíncronas)
- ↘ Java Management Extensions (JMX) (administração da sua aplicação e estatísticas sobre a mesma)

Especificações: JSP e Servlets

(mesmo usando frameworks e bibliotecas)

São especificações essenciais que todo desenvolvedor Java vai precisar para desenvolver com a Web.

Servidor de Aplicação

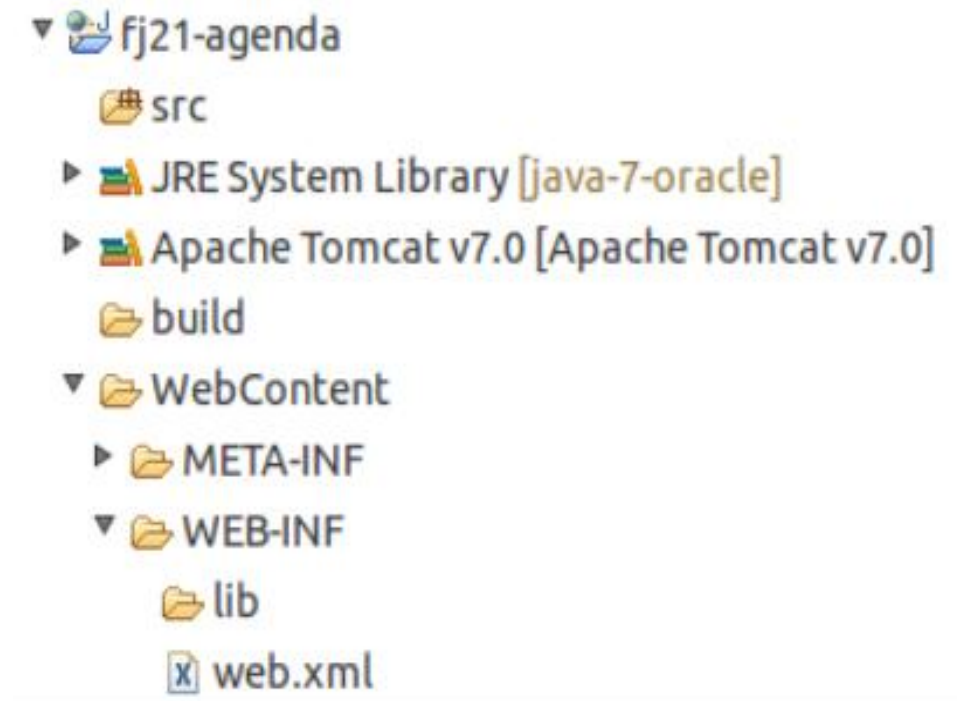
- ↪ Implementa a especificação J2EE
- ↪ Existem diversas implementações (exemplo):
 - Oracle/Sun, GlassFish Server Open Source Edition 4.0, gratuito;
 - RedHat, JBoss Application Server 7.x, gratuito, Java EE 6;
 - Apache, Apache Geronimo, gratuito, Java EE 6;
 - Oracle/BEA, Oracle WebLogic Server 8.x, Java EE 6;
 - IBM, IBM WebSphere Application Server, Java EE 6;
 - SAP, SAP NetWeaver Application Server ou SAP Web Application Server, Java EE 6 Web Profile;
- ↪ Alguns servidores não implementam toda a J2EE (“*Servlet Containers*”), p.ex.:
 - Tomcat
 - Jetty
- ↪ Vamos utilizar principalmente o Tomcat.

Preparando o IDE

- Tomcat como serviço independente
- Tomcat integrado ao Eclipse

Primeiro Projeto Web

Estrutura de Pastas e Arquivos



Resumo das Pastas

- ℓ **src** - códigos fonte Java (.java). Estrutura de pacotes que organizam o código fonte.
- ℓ **build** - onde o Eclipse compila as classes (.class)
- ℓ **WebContent** - content directory (páginas, imagens, css etc)
 - ℓ Conteúdo acessível na URL do projeto. Estrutura de pacotes de páginas (html e jsp) e arquivos como (javascript e css).
- ℓ **WebContent/WEB-INF/** - pasta oculta com configurações e recursos do projeto
- ℓ **WebContent/WEB-INF/lib/** - bibliotecas .jar
- ℓ **WebContent/WEB-INF/classes/** - arquivos compilados (da pasta build) são copiados para cá.

Referências Bibliográficas

BASHAM, Bryan; SIERRA, KATHY; Bates, Bert.
Use a Cabeça: Servlets & JSP. O'Reilly. 2011.

CAELUM. Java para Desenvolvimento Web. Curso
FJ-21.