

# Capítulo 3



Java JSP



- Que es un JSP?
  - JavaServer Pages
  - Código Java en un documento HTML
- Es una solución para dos problemas
  - Integración entre diseñadores y desarrolladores de aplicaciones
    - Los diseñadores podrían aprender solo la sintaxis JSP
  - Formatear HTML en un String no es muy elegante
    - Las reglas de un String pueden hacer complejo el uso de caracteres especiales de HTML
      - Saltos de línea
      - Comillas



- Cuando un cliente accede a la URL del JSP
  - El código Java se ejecuta en el servidor, generando datos
  - Los datos generados, rodeados de “tags” HTML son enviados al cliente
  - El cliente no conoce el código JSP, solo conoce el resultado de su ejecución
- “Server-side include”
  - Nombre de la metodología que combina lenguajes de programación con HTML
  - El lenguaje de programación usado se conoce como “scripting language”



- JavaServer Pages (JSP)
  - Es la tecnología que provee una **especificación** estándar para combinar Java con HTML
  - Especificación JSP
    - enumera la sintaxis y describe la semántica de los elementos que componen una página JSP
    - Los elementos se llaman JSP tags
- Ejemplo de uso
  - Hello.html, HelloServlet.java y Hello.jsp



- Servlet o JSP?
  - Servlet y JSP pueden hacer lo mismo?
  - Recordar que los Servlets son extensiones del servidor y proveen funcionalidades extras
    - Implementación de servicios especiales: validaciones con bases de datos, transacciones, autenticaciones, etc
  - Servlet actúa como un controlador de componentes
    - Controla la lógica de negocio
    - Son desarrollados por programadores Java, que tienen conocimientos de OOP
  - JSP son páginas web, similares en estructura a HTML
    - Un diseñador de páginas web con conocimientos de tags JSP y un poco de Java puede escribir páginas JSP



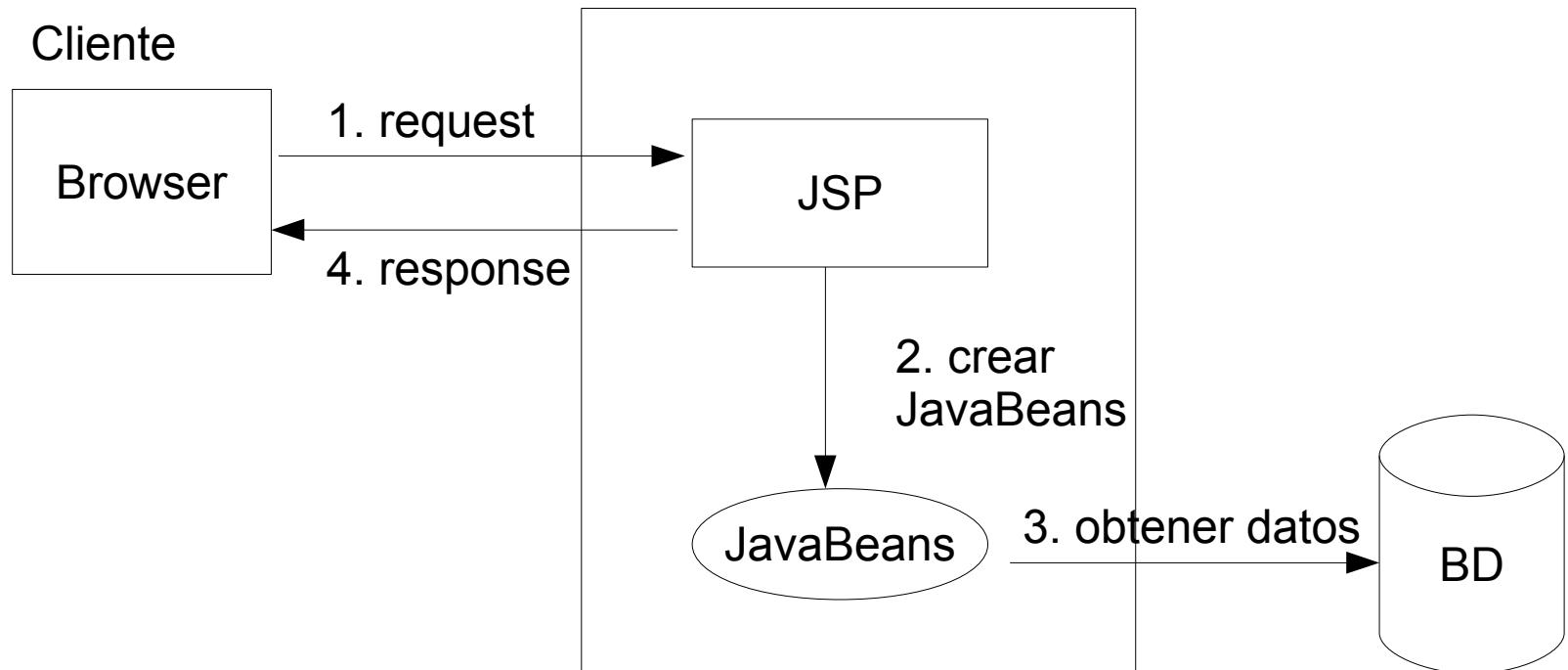
- Las aplicaciones WEB típicamente combinan páginas JSP y Servlets
- Ejemplo: Proceso de autenticación de usuarios
  - El código que genera un formulario HTML, mensajes de éxito o error, deberían ir en un JSP
  - El código que accede a una base de datos, valida la password y autentica al usuario debería ir en un servlet
- Mantener siempre en mente
  - Las páginas JSP se enfocan en presentación visual
  - La lógica de negocio de la aplicación se deja para los servlets



- Modelos de arquitectura
- Modelo 1
  - Todos los requerimientos web son atendidos por páginas JSP
    - Incluyendo autenticaciones de cliente, acceso a bases de datos, etc
  - Es adecuada para aplicaciones simples
  - Inconvenientes para usar en aplicaciones complejas
    - Lógica de negocio en trozos de código Java en el JSP
    - No se aplica la reusabilidad de componentes
      - El código de autenticación de usuarios escrito en un JSP no se puede reutilizar por otros JSP



- Modelo 1







- Modelo 2
  - Sigue el modelo del patrón de diseño MVC: Model-View-Controller
  - Todos los requerimientos son atendidos por un servlet que **controla** la aplicación
    - Analiza los requerimientos y obtiene los datos necesarios para generar la respuesta en objetos JavaBeans
    - Objetos JavaBeans son el **modelo** de la aplicación
    - Finalmente el controlador envía el requerimiento a una página JSP
    - Está página utiliza los datos almacenados en el JavaBean para generar la presentación de la respuesta
    - Las páginas JSP forma la **vista** de la aplicación
  - La gran ventaja de este modelo es la separación de responsabilidades



- Modelo 2
  - El controlador es un único punto de entrada a la aplicación
    - Puede usar componentes de seguridad o control de estados
    - Estas componentes se pueden reutilizar
  - Dependiendo del requerimiento del cliente
    - el controlador lo enviará a la componente de presentación adecuada
    - Esta componente generará la respuesta al usuario
  - Los JSP quedan sin lógica de negocio compleja
    - Se dedican solo a la presentación de datos



- Modelo 2

