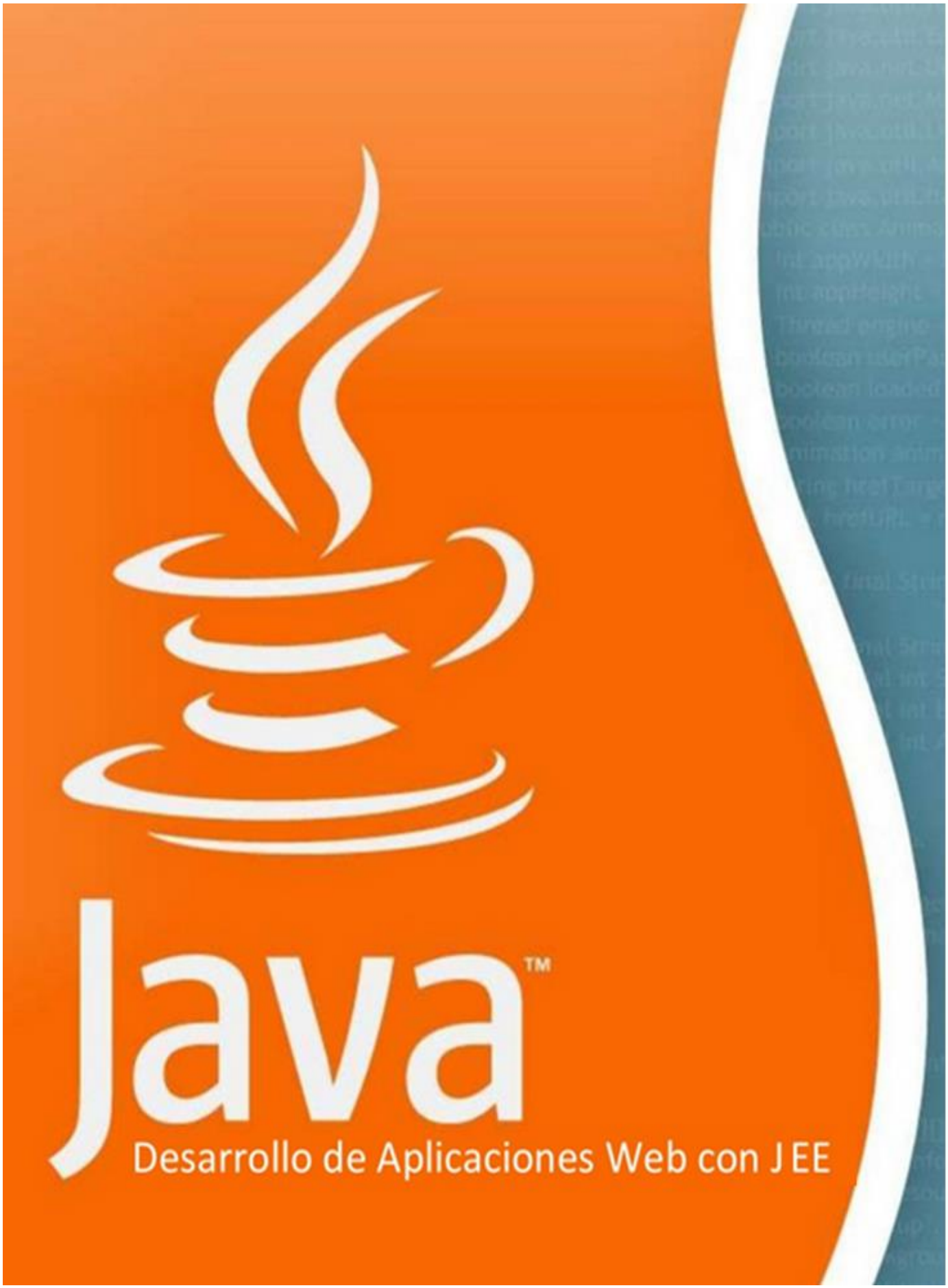

Java™
Desarrollo de Aplicaciones Web con J EE

PARTE IV

COMUNICAR SERVLETS

3.9 Comunicar procesos entre Servlets. Reenvío peticiones.

Los Servlets pueden delegar peticiones a otros Servlets/JSP/HTML. Hay dos formas de redirigir una petición a otro recurso:

- **Redirecciones HTTP (sendRedirect):**

  - ✓ El servidor envía una respuesta al cliente y la URI a la que este debe enviar la petición
  - ✓ No permite enviar los parámetros de la petición
  - ✓ Permite redireccionar a URLs externas a la aplicación web

- **Redirecciones internas en el servidor (RequestDispatcher):**

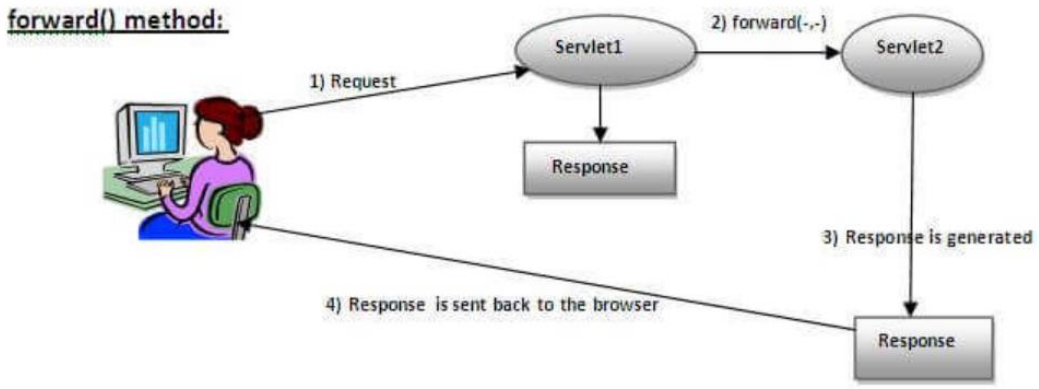
  - ✓ Es más rápido que sendRedirect, ya que se trata a nivel de servidor
  - ✓ Se redirige la petición de un recurso a otro dentro de la misma aplicación Web
  - ✓ El recurso de la última redirección devuelve al cliente la respuesta HTTP
  - ✓ La redirección es transparente para el cliente. El cliente no se entera de la redirección (p.e., el navegador muestra la URI original de la petición, no la redirigida)
  - ✓ El control retorna al finalizar el método del despachador, por lo que conviene que sea lo último que se ejecuta
  - ✓ Permite enviar los parámetros de la petición

La clase ***jakarta.servlet.RequestDispatcher*** se encargará de reenviar peticiones a otros servlets, jsp o htmls a través del método **forward** o el método **include**. Ambos métodos reciben dos parámetros, el request y el response. Para ello:

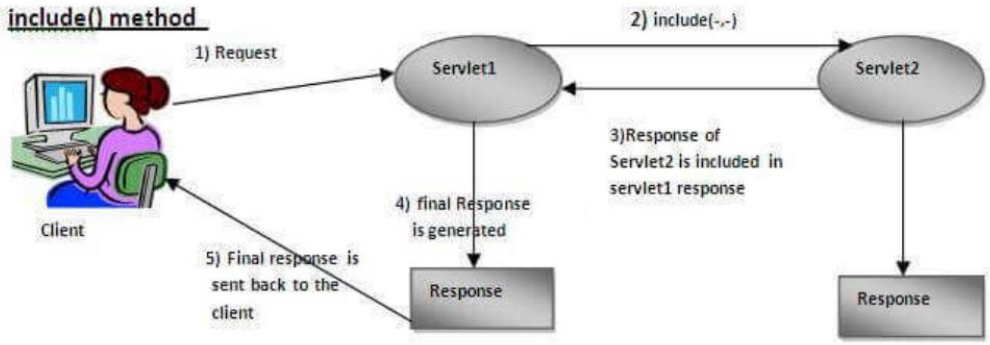
- Se invoca el método g**etRequestDispacher**(rutaRecursoAlQueSeDelegaelDespachador)
- Redirige la respuesta al recurso mediante el método:

✓ **fordward(request,response) :** Permite a un servlet procesar una petición parcialmente y luego pasar la petición a otro servlet para generar la respuesta final. Es muy importante saber que **cualquier línea enviada al objeto out antes o después de la ejecución del método no tendrá efecto**, sí las del objeto que recibe el reenvío. Cualquier otro tipo de línea sí se ejecutará.

Además, tiene la particularidad de que **conserva los parámetros de petición** y los pone a disposición del servlet/JSP/HTML a reenviar, no ocurre así con sendRedirect que los pierde.



- ✓ **include (request,response):** Permite incluir el contenido de otro recurso en la respuesta que está generando el recurso que invoca el método, no tiene el mismo efecto que el forward**, lo que se mande al objeto out siempre se verá.** su objetivo es el de escribir parte del documento y delegar después el resto a otro servlet.



3.9.1 Ejemplo 10

```
/**
 * Servlet implementation class Ejemplo10
 */
public class Ejemplo10 extends HttpServlet {
    private static final long serialVersionUID = 1L;

    /**
     * @see HttpServlet#HttpServlet()
     */
    public Ejemplo10() {
        super();
        // TODO Auto-generated constructor stub
    }


    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

        response.setContentType("text/html");
        response.getWriter().append("<html><body>")
                            .append("<form name='formulario' action='Ejemplo10' method='POST'>")
                            .append("<input type='submit' name='metodo' value='include'  />")
                            .append("<input type='submit' name='metodo' value='forward' />")
                            .append("</form></body></html>");
    }

    /**{
}
     * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)
     */
    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        response.setContentType("text/html");
        response.getWriter().append("<html>")
                            .append("<body style='background-color:#AAFF9F'>")
                            .append("<H3>Ejemplo de RequestDispatcher</H3>")
                            .append("<p>Este servlet usar  un despachador que nos lleve a un servlet despachador </p>");
        //Rutas absolutas, no permite rutas relativas
        //RequestDispatcher dispatcher = sc.getRequestDispatcher("/Ejemplo10b");
        //Rutas relativas y absolutas pero solo para el contexto de la app web
        RequestDispatcher dispatcher = request.getRequestDispatcher("Ejemplo10b");
        if (dispatcher != null) {
            if (request.getParameter("metodo") != null) {
                request.setAttribute("attribDitpach", "Llega atributo");
                if (request.getParameter("metodo").equalsIgnoreCase("include")) {
                    dispatcher.include(request, response);
                } else { // forward
                    dispatcher.forward(request, response);
                }
            }
            response.getWriter().append("<p>Final del servlet despachador</p>");
        } else {
            response.getWriter().append("<p>No se ha encontrado el despachador</p>");
        }
        response.getWriter().append("</body></html>");
        response.getWriter().close();
    }

}
```

```
public class Ejemplo10b extends HttpServlet {
    private static final long serialVersionUID = 1L;

    /**
     * @see HttpServlet#HttpServlet()
     */
    public Ejemplo10b() {
        super();
        // TODO Auto-generated constructor stub
    }


    /**
     * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)
     */
    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        response.setContentType("text/html");
        response.getWriter().append("<html><body>")
        .append("<body>")
        .append("<p>Inicio del dispatcher</p>")
        .append("<h1>HOLA DESPACHADOR</h1>")
        .append((String)request.getAttribute("attribDitpach"))
        .append("<p>Fin de dispatcher</p>")
        .append("</body>")
        .append("</html>");
        //out.close();
    }

}
```