



escuela técnica superior
de ingeniería informática



Introducción a Java EE: Servlets

Departamento de
Lenguajes y Sistemas Informáticos

Grupo de Ingeniería del Software

Marzo 2007

UNIVERSIDAD DE SEVILLA


[Ángel US V7] Diseño: Amador Durán Toro (2003-2006)

Versión original: Amador Durán y Ramon Medel (marzo 2007)




Escuela Técnica Superior
de Ingeniería Informática
Departamento de Lenguajes
y Sistemas Informáticos

1. ¿Qué es un servlet?
2. Ciclo de vida de un servlet
3. Ejemplo de un servlet simple
4. Datos de formularios
5. El objeto *response*
6. Objetos de ámbito
7. Reenvío y redireccionamiento
8. Filtros y gestores de eventos
9. Despliegue de un servlet
10. Desarrollo con IDEs
11. Para saber más
12. Ejercicios



Introducción a Java EE: Servlets

- ¿Qué es un *servlet*?
 - Un *servlet* es un objeto Java que se ejecuta dentro de un *contenedor de servlets* (p.e. Tomcat) ampliando su funcionalidad.
 - Un *servlet* implementa la interfaz `javax.servlet.Servlet` y, normalmente, es un objeto de una clase derivada de `javax.servlet.http.HttpServlet`, ya que se suele trabajar sobre el protocolo HTTP.



request →

← response

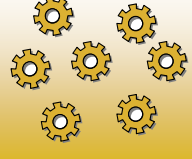
Servidor HTTP

HTML

CSS

XML

Contenedor de *servlets*



Versión original: Amador Durán y Ramon Medel (febrero 2007)


Sevilla, marzo de 2007
Grupo de Ingeniería del Software

UNIVERSIDAD DE SEVILLA

1


1

[Angel Luis V7] Diseño: Amador Durán Toro (2003-2006)



Escuela Técnica Superior de Ingeniería Informática
Departamento de Lenguajes y Sistemas Informáticos

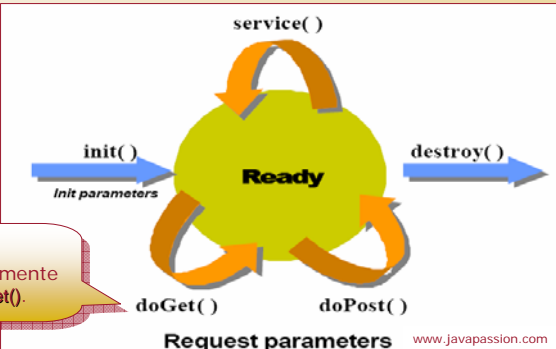
1. ¿Qué es un servlet?
2. Ciclo de vida de un servlet
3. Ejemplo de un servlet simple
4. Datos de formularios
5. El objeto *response*
6. Objetos de ámbito
7. Reenvío y redireccionamiento
8. Filtros y gestores de eventos
9. Despliegue de un servlet
10. Desarrollo con IDEs
11. Para saber más
12. Ejercicios



Sevilla, marzo de 2007
Grupo de Ingeniería del Software

Introducción a Java EE: Servlets

- La vida de un *servlet*
 - El contenedor crea tantos *servlets* como considere necesario y los usa concurrentemente.
 - Los inicializa con `init()`, los usa con `service()` (que llama a `doPost()` o `doGet()`) y, cuando lo considera oportuno, los destruye llamando antes a `destroy()`.



www.javapassion.com

Lo habitual es redefinir únicamente `doPost()` y `doGet()`.

Versión original: Amador Durán y Ramon Medel (febrero 2007)

2

[Angel Luis V7] Diseño: Amador Durán Toro (2003-2006)



Escuela Técnica Superior de Ingeniería Informática
Departamento de Lenguajes y Sistemas Informáticos

1. ¿Qué es un servlet?
2. Ciclo de vida de un servlet
3. Ejemplo de un servlet simple
4. Datos de formularios
5. El objeto *response*
6. Objetos de ámbito
7. Reenvío y redireccionamiento
8. Filtros y gestores de eventos
9. Despliegue de un servlet
10. Desarrollo con IDEs
11. Para saber más
12. Ejercicios



Sevilla, marzo de 2007
Grupo de Ingeniería del Software

Introducción a Java EE: Servlets

- Un *servlet* sencillo

```

public class MiServlet extends HttpServlet {
    public void doGet( HttpServletRequest request,
                      HttpServletResponse response )
                      throws ServletException, IOException
    {
        PrintWriter out = response.getWriter();
        response.setContentType("text/html");
        out.println("<html>");
        out.println("<head><title>Ejemplo</title></head>");
        out.println("<body><h1>¡Hola, mundo!</h1></body>");
        out.println("</html>");
        out.close();
    }

    public void doPost( ... ) throws ...
    {
        doGet( request, response );
    }
}

```

Versión original: Amador Durán y Ramon Medel (febrero 2007)

3

(Angel US V7) Diseño: Amador Durán Toro (2003-2006)



Escuela Técnica Superior de Ingeniería Informática
Departamento de Lenguajes y Sistemas Informáticos

1. ¿Qué es un *servlet*?
2. Ciclo de vida de un *servlet*
3. Ejemplo de un *servlet* simple
4. Datos de formularios
5. El objeto *response*
6. Objetos de ámbito
7. Reenvío y redireccionamiento
8. Filtros y gestores de eventos
9. Despliegue de un *servlet*
10. Desarrollo con IDEs
11. Para saber más
12. Ejercicios



Sevilla, marzo de 2007
Grupo de Ingeniería del Software

Introducción a Java EE: Servlets

- Acceso a datos de un formulario
 - Un *servlet* puede usarse para procesar un formulario mediante el atributo *action* del elemento `<form>`.
 - Para acceder a los datos enviados por el formulario se usa el método `getParameter("nombre")` del objeto *request*, que devuelve una cadena con el valor o *null* si no existe el valor. Para parámetros multivaluados se debe usar `getParameterValues("nombre")`.
 - No se incluyen métodos para subir archivos*.

```
public void doGet( HttpServletRequest request,
                  HttpServletResponse response ) throws ...
{
    ...
    String nombre = request.getParameter("nombre");
    String apellidos = request.getParameter("apellidos");
    Integer edad = Integer.parseInt( request.getParameter("edad") );
    ...
}
```

*Existe software libre que lo hace.



4

(Angel US V7) Diseño: Amador Durán Toro (2003-2006)



Escuela Técnica Superior de Ingeniería Informática
Departamento de Lenguajes y Sistemas Informáticos

1. ¿Qué es un *servlet*?
2. Ciclo de vida de un *servlet*
3. Ejemplo de un *servlet* simple
4. Datos de formularios
5. El objeto *response*
6. Objetos de ámbito
7. Reenvío y redireccionamiento
8. Filtros y gestores de eventos
9. Despliegue de un *servlet*
10. Desarrollo con IDEs
11. Para saber más
12. Ejercicios



Sevilla, marzo de 2007
Grupo de Ingeniería del Software

Introducción a Java EE: Servlets

- El objeto *response* (`HttpServletResponse`)
 - Representa a una respuesta del protocolo HTTP, incluyendo el estado, las cabeceras y el cuerpo.
 - Permite enviar tanto texto (`getWriter()`) como datos binarios (`getOutputStream()`).
 - Permite también dar valor a *cookies* con `addCookie()`.

```
public void doGet( HttpServletRequest request,
                  HttpServletResponse response ) throws ...
{
    PrintWriter out = response.getWriter();


    response.setContentType("text/html");
    out.println("<html>");
    ...
    out.println("</html>");
    out.close();
}
```

*Existe software libre que lo hace.



5

[Angel US V7] Diseño: Amador Durán Toro (2003-2006)



Escuela Técnica Superior de Ingeniería Informática
Departamento de Lenguajes y Sistemas Informáticos

1. ¿Qué es un *servlet*?
2. Ciclo de vida de un *servlet*
3. Ejemplo de un *servlet* simple
4. Datos de formularios
5. El objeto *response*
6. Objetos de ámbito
7. Reenvío y redireccionamiento
8. Filtros y gestores de eventos
9. Despliegue de un *servlet*
10. Desarrollo con IDEs
11. Para saber más
12. Ejercicios




Sevilla, marzo de 2007
Grupo de Ingeniería del Software

Introducción a Java EE: Servlets

- ¿Qué son los objetos de ámbito?
 - Los objetos de ámbito contienen un diccionario (*array* asociativo) de objetos Java a los que se accede con los métodos:
 - `getAttribute("nombre")` → objeto
 - `setAttribute("nombre", objeto)`
- Existen tres tipos de objetos de ámbito*:
 - Contexto web: un solo objeto por aplicación
 - Sesión: un objeto por *sesión*
 - Petición (*request*): un objeto por petición HTTP
- Ciclos de vida
 - Cada objeto de ámbito tiene un ciclo de vida diferente, asociado al concepto asociado:
 - 1 aplicación
 - *n* sesiones por aplicación
 - *m* peticiones por sesión

*Realmente son cuatro si se incluyen las páginas JSP.



Versión original: Amador Durán y Ramon Medel (febrero 2007)

6

[Angel US V7] Diseño: Amador Durán Toro (2003-2006)



Escuela Técnica Superior de Ingeniería Informática
Departamento de Lenguajes y Sistemas Informáticos

1. ¿Qué es un *servlet*?
2. Ciclo de vida de un *servlet*
3. Ejemplo de un *servlet* simple
4. Datos de formularios
5. El objeto *response*
6. Objetos de ámbito
7. Reenvío y redireccionamiento
8. Filtros y gestores de eventos
9. Despliegue de un *servlet*
10. Desarrollo con IDEs
11. Para saber más
12. Ejercicios



Sevilla, marzo de 2007
Grupo de Ingeniería del Software

Introducción a Java EE: Servlets

- Contexto web (aplicación)
 - Es un único objeto compartido con un ámbito correspondiente a toda la aplicación web (contexto).
 - Permite escribir mensajes en el *log* del servidor.
 - También se usa para acceder al objeto que permite reenviar y redireccionar peticiones*.

```

public void doGet( HttpServletRequest request,
                  HttpServletResponse response ) throws ...
{
    ServletContext contexto = getServletContext();
    ...
    contexto.log("Mensaje para el log");
    ...
    contexto.setAttribute("nombre1", objeto1);
    ...
    otroObjeto = (ClaseObjeto)contexto.getAttribute("objeto2");
}
            
```

*Se verá en el siguiente apartado.



Versión original: Amador Durán y Ramon Medel (febrero 2007)

7

[Angel US V7] Diseño: Amador Durán Toro (2003-2006)



Escuela Técnica Superior de Ingeniería Informática
Departamento de Lenguajes y Sistemas Informáticos

1. ¿Qué es un *servlet*?
2. Ciclo de vida de un *servlet*
3. Ejemplo de un *servlet* simple
4. Datos de formularios
5. El objeto *response*
6. Objetos de ámbito
7. Reenvío y redireccionamiento
8. Filtros y gestores de eventos
9. Despliegue de un *servlet*
10. Desarrollo con IDEs
11. Para saber más
12. Ejercicios



Sevilla, marzo de 2007
Grupo de Ingeniería del Software

Introducción a Java EE: Servlets


- Sesión
 - Son objetos asociados a las *sesiones* de los usuarios. Cada sesión tiene un objeto propio.
 - Permite guardar información sobre el *login*, un *carrito de la compra*, la página actual de un listado paginado, etc.

```
public void doGet( HttpServletRequest request,
                  HttpServletResponse response ) throws ...
{
    HttpSession session = request.getSession();
    ...
    Carrito carrito = (Carrito)session.getAttribute("carrito");
    ...
    session.setAttribute("paginaActual", pagina);
}
```

Versión original: Amador Durán y Ramon Medel (febrero 2007)


8

[Angel US V7] Diseño: Amador Durán Toro (2003-2006)



Escuela Técnica Superior de Ingeniería Informática
Departamento de Lenguajes y Sistemas Informáticos

1. ¿Qué es un *servlet*?
2. Ciclo de vida de un *servlet*
3. Ejemplo de un *servlet* simple
4. Datos de formularios
5. El objeto *response*
6. Objetos de ámbito
7. Reenvío y redireccionamiento
8. Filtros y gestores de eventos
9. Despliegue de un *servlet*
10. Desarrollo con IDEs
11. Para saber más
12. Ejercicios




Sevilla, marzo de 2007
Grupo de Ingeniería del Software

Introducción a Java EE: Servlets

- Petición (request)
 - Representa una petición del protocolo HTTP.
 - Contiene los datos enviados desde formularios*, las *cookies* y un diccionario de atributos como el del contexto y la sesión.
 - Contiene información del navegador que hace la petición y del usuario si se ha autenticado mediante el contenedor de *servlets*.

```
public void doGet( HttpServletRequest request,
                  HttpServletResponse response ) throws ...
{
    ... request.getLocale(); ... request.getRemoteAddr();
    ... request.getServerName(); ... request.getServerPort();
    ... request.getContextPath(); ... request.getRemoteUser();
    ... If ( request.isUserInRole("admin") ) { ... }
    Cookie[] cookies = request.getCookies();
    ...
}
```

Versión original: Amador Durán y Ramon Medel (febrero 2007)



*Se vio en el apartado anterior.

9

[Angel US V7] Diseño: Amador Durán Toro (2003-2006)



Escuela Técnica Superior de Ingeniería Informática
Departamento de Lenguajes y Sistemas Informáticos

1. ¿Qué es un *servlet*?
2. Ciclo de vida de un *servlet*
3. Ejemplo de un *servlet* simple
4. Datos de formularios
5. El objeto *response*
6. Objetos de ámbito
7. Reenvío y redireccionamiento
8. Filtros y gestores de eventos
9. Despliegue de un *servlet*
10. Desarrollo con IDEs
11. Para saber más
12. Ejercicios



Sevilla, marzo de 2007
Grupo de Ingeniería del Software

Introducción a Java EE: Servlets

- Reenvío desde un *servlet* (*forwarding*)
 - A veces se hace cierto procesamiento en un *servlet* y a continuación se invoca otro *servlet* o página JSP.
 - Para ello es necesario usar un objeto de la clase *RequestDispatcher* obtenido a través de *request*.
 - Si es necesario pasar información se puede pasar dentro del diccionario de atributos de *request*.

```
public void doGet( HttpServletRequest request,
                  HttpServletResponse response ) throws ...
{
    /* procesamiento previo */
    /* request.setAttribute( "nombre", objeto ); */

    RequestDispatcher dispatcher =
        request.getRequestDispatcher("/otraPagina.jsp");
    if ( dispatcher != null )
        dispatcher.forward( request, response );
}
```

Versión original: Amador Durán y Ramon Medel (febrero 2007)


10

[Angel US V7] Diseño: Amador Durán Toro (2003-2006)



Escuela Técnica Superior de Ingeniería Informática
Departamento de Lenguajes y Sistemas Informáticos

1. ¿Qué es un *servlet*?
2. Ciclo de vida de un *servlet*
3. Ejemplo de un *servlet* simple
4. Datos de formularios
5. El objeto *response*
6. Objetos de ámbito
7. Reenvío y redireccionamiento
8. Filtros y gestores de eventos
9. Despliegue de un *servlet*
10. Desarrollo con IDEs
11. Para saber más
12. Ejercicios



Sevilla, marzo de 2007
Grupo de Ingeniería del Software

Introducción a Java EE: Servlets

- Redireccionamiento desde un *servlet*
 - En otras ocasiones, lo único que se desea es redireccionar una petición hacia otro recurso.
 - A diferencia del *forwarding*, el usuario observa que la URL cambia al redireccionarlo.
 - Se hace mediante el método *sendRedirect()* de *response*.

```
public void doGet( HttpServletRequest request,
                  HttpServletResponse response ) throws ...
{
    ...
    if ( ... )
        response.sendRedirect("/otraPagina.jsp");
    else
        response.sendRedirect("/otroServlet");
    ...
}
```

Versión original: Amador Durán y Ramon Medel (febrero 2007)


11

UNIVERSIDAD DE SEVILLA
Escuela Técnica Superior de Ingeniería Informática
Departamento de Lenguajes y Sistemas Informáticos

1. ¿Qué es un *servlet*?
2. Ciclo de vida de un *servlet*
3. Ejemplo de un *servlet* simple
4. Datos de formularios
5. El objeto *response*
6. Objetos de ámbito
7. Reenvío y redireccionamiento
8. Filtros y gestores de eventos
9. Despliegue de un *servlet*
10. Desarrollo con IDEs
11. Para saber más
12. Ejercicios



Sevilla, marzo de 2007
Grupo de Ingeniería del Software

Introducción a Java EE: Servlets

- Filtros para *servlets*
 - Los filtros son objetos Java que pueden añadir funcionalidad a los *servlets* (u otros artefactos web) interceptando las peticiones y las respuestas.
 - Es una implementación del patrón de diseño *Decorador*.
 - Permiten gestionar el control de acceso, registros (*logs*), transformar contenido, mantener una caché de páginas dinámicas, etc.
 - Pueden encadenarse y formar una *cadena de filtros*.
 - Se configuran en *web.xml*



Versión original: Amador Durán y Ramon Medel (febrero 2007)

UNIVERSIDAD DE SEVILLA

12

UNIVERSIDAD DE SEVILLA
Escuela Técnica Superior de Ingeniería Informática
Departamento de Lenguajes y Sistemas Informáticos

1. ¿Qué es un *servlet*?
2. Ciclo de vida de un *servlet*
3. Ejemplo de un *servlet* simple
4. Datos de formularios
5. El objeto *response*
6. Objetos de ámbito
7. Reenvío y redireccionamiento
8. Filtros y gestores de eventos
9. Despliegue de un *servlet*
10. Desarrollo con IDEs
11. Para saber más
12. Ejercicios



Sevilla, marzo de 2007
Grupo de Ingeniería del Software

Introducción a Java EE: Servlets

- Gestores de eventos (*listeners*) para *servlets*
 - Permiten gestionar los siguientes eventos:
 - Inicio del contexto (aplicación)
 - Finalización del contexto (aplicación)
 - Cambios en los atributos del contexto (aplicación)
 - Inicio de una sesión
 - Invalidación (fin) de una sesión
 - Cambios en los atributos de sesión
 - Son una implementación del patrón de diseño *Observador*.
 - Se configuran en *web.xml*



Versión original: Amador Durán y Ramon Medel (febrero 2007)

UNIVERSIDAD DE SEVILLA

13

(Angel US VTI Diener, Amador Durán Toro (2003-2006))



Escuela Técnica Superior
de Ingeniería Informática
Departamento de Lenguajes
y Sistemas Informáticos

1. ¿Qué es un servlet?
2. Ciclo de vida de un servlet
3. Ejemplo de un servlet simple
4. Datos de formularios
5. El objeto *response*
6. Objetos de ámbito
7. Reenvío y redireccionamiento
8. Filtros y gestores de eventos
9. Despliegue de un servlet
10. Desarrollo con IDEs
11. Para saber más
12. Ejercicios



Sevilla, marzo de 2007
Grupo de Ingeniería del Software

Introducción a Java EE: Servlets

- Copiar el archivo compilado en la aplicación
 - Copiar el .class en WEB-INF/classes.
- Elementos en *web.xml*
 - Para cada *servlet* que se despliega en una aplicación web debe haber un elemento `<servlet>` y uno o más elementos `<servlet-mapping>` en *web.xml*.
- Elemento `<servlet>`
 - Los elementos hijos habituales son los siguientes:
 - `<description>`Descripción opcional del servlet`</description>`
 - `<servlet-name>`Nombre único del servlet`</servlet-name>`
 - `<servlet-class>`clase del servlet`</servlet-class>`
- Elemento `<servlet-mapping>`
 - Los elementos hijos habituales son los siguientes:
 - `<servlet-name>`Servlet que se mapea`</servlet-name>`
 - `<url-mapping>``/URLParaElServlet</url-mapping>`

<http://servidor:puerto/contexto/URLParaElServlet>


Versión original: Amador Durán y Ramon Medel (febrero 2007)

(Angel US VTI Diener, Amador Durán Toro (2003-2006))



Escuela Técnica Superior
de Ingeniería Informática
Departamento de Lenguajes
y Sistemas Informáticos

1. ¿Qué es un servlet?
2. Ciclo de vida de un servlet
3. Ejemplo de un servlet simple
4. Datos de formularios
5. El objeto *response*
6. Objetos de ámbito
7. Reenvío y redireccionamiento
8. Filtros y gestores de eventos
9. Despliegue de un servlet
10. Desarrollo con IDEs
11. Para saber más
12. Ejercicios



Sevilla, marzo de 2007
Grupo de Ingeniería del Software

Introducción a Java EE: Servlets

- Ejemplo de despliegue

```

<web-app>
...
<servlet>
  <description>Servlet de ejemplo de despliegue</description>
  <servlet-name>ServletEjemploJavaEE</servlet-name>
  <servlet-class>es.us.isi.ServletEjemplo</servlet-class>
...
</servlet>
...
<servlet-mapping>
  <servlet-name>ServletEjemploJavaEE</servlet-name>
  <url-mapping>/Ejemplo</url-mapping>
</servlet-mapping>
...
</web-app>
            
```

Versión original: Amador Durán y Ramon Medel (febrero 2007)

[Angel US V7] Diseño: Amador Durán Toro (2003-2006)



Escuela Técnica Superior de Ingeniería Informática
Departamento de Lenguajes y Sistemas Informáticos

1. ¿Qué es un *servlet*?
2. Ciclo de vida de un *servlet*
3. Ejemplo de un *servlet* simple
4. Datos de formularios
5. El objeto *response*
6. Objetos de ámbito
7. Reenvío y redireccionamiento
8. Filtros y gestores de eventos
9. Despliegue de un *servlet*
10. Desarrollo con IDEs
11. Para saber más
12. Ejercicios



Sevilla, marzo de 2007
Grupo de Ingeniería del Software

Introducción a Java EE: Servlets

- Creación de un *servlet* en Eclipse 3.2



Versión original: Amador Durán y Ramon Medel (febrero 2007)


16

[Angel US V7] Diseño: Amador Durán Toro (2003-2006)



Escuela Técnica Superior de Ingeniería Informática
Departamento de Lenguajes y Sistemas Informáticos

1. ¿Qué es un *servlet*?
2. Ciclo de vida de un *servlet*
3. Ejemplo de un *servlet* simple
4. Datos de formularios
5. El objeto *response*
6. Objetos de ámbito
7. Reenvío y redireccionamiento
8. Filtros y gestores de eventos
9. Despliegue de un *servlet*
10. Desarrollo con IDEs
11. Para saber más
12. Ejercicios



Sevilla, marzo de 2007
Grupo de Ingeniería del Software

Introducción a Java EE: Servlets

- Creación de un *servlet* en NetBeans 5.5



Versión original: Amador Durán y Ramon Medel (febrero 2007)


17

[Angel US VTI Diseño: Amador Durán Toro (2003-2006)]



Escuela Técnica Superior de Ingeniería Informática
Departamento de Lenguajes y Sistemas Informáticos

1. ¿Qué es un *servlet*?
2. Ciclo de vida de un *servlet*
3. Ejemplo de un *servlet* simple
4. Datos de formularios
5. El objeto *response*
6. Objetos de ámbito
7. Reenvío y redireccionamiento
8. Filtros y gestores de eventos
9. Despliegue de un *servlet*
10. Desarrollo con IDEs
11. Para saber más
12. Ejercicios



Sevilla, marzo de 2007
Grupo de Ingeniería del Software

Introducción a Java EE: Servlets

- Bibliografía recomendada
 - *Servlet Basics*. S. Ching.
<http://www.javapassion.com>, 2006.
 - *Servlet Advanced*. S. Ching.
<http://www.javapassion.com>, 2006.
 - *Core Servlets and JavaServer Pages (1st ed.)*. M. Hall.
<http://pdf.coreservlets.com>, 2001.







Versión original: Amador Durán y Ramon Medel (febrero 2007)

[Angel US VTI Diseño: Amador Durán Toro (2003-2006)]



Escuela Técnica Superior de Ingeniería Informática
Departamento de Lenguajes y Sistemas Informáticos

1. ¿Qué es un *servlet*?
2. Ciclo de vida de un *servlet*
3. Ejemplo de un *servlet* simple
4. Datos de formularios
5. El objeto *response*
6. Objetos de ámbito
7. Reenvío y redireccionamiento
8. Filtros y gestores de eventos
9. Despliegue de un *servlet*
10. Desarrollo con IDEs
11. Para saber más
12. Ejercicios



Sevilla, marzo de 2007
Grupo de Ingeniería del Software

Introducción a Java EE: Servlets

- Desarrollar *servlets*
 - Escriba algunos ejemplos de *servlets* sencillos:
 - El clásico ejemplo del "¡Hola, mundo!".
 - Uno que reciba datos de un formulario y genere una respuesta con esos datos.
 - Uno que reciba datos de un formulario, cree un objeto Java con esa información y la pase a través de un objeto de ámbito a otro *servlet* mediante *forwarding*.
- Investigar sobre la recepción de archivos
 - Recibir archivos a través de *request* no es posible con la especificación actual del API de *servlets*. Investigue alternativas de software libre para solucionar este problema.
- Investigar sobre filtros y *listeners*
 - Busque información sobre cómo desarrollar filtros y *listeners* y sus posibles aplicaciones.
- Usar Ant para compilar y desplegar un *servlet*
 - Aunque los IDEs actuales despliegan automáticamente los *servlets*, conviene saber hacerlo con una herramienta como *Ant*. Investigue sobre ello.



Versión original: Amador Durán y Ramon Medel (febrero 2007)