

Novática, revista fundada en 1975 y decana de la prensa informática española, es el órgano oficial de expresión y formación continua de **ATI** (Asociación de Técnicos de Informática), organización que edita también la revista **REICIS** (Revista Española de Innovación, Calidad e Ingeniería del Software). **Novática** edita asimismo **UPGRADE**, revista digital de **CEPIS** (Council of European Professional Informatics Societies), en lengua inglesa, y es miembro fundador de **UPENET** (UPGRADE European Network).

<<http://www.ati.es/novatica/>>
 <<http://www.ati.es/reicis/>>
 <<http://www.upgrade-cepis.org/>>

ATI es miembro fundador de **CEPIS** (Council of European Professional Informatics Societies) y es representante de España en **IFIP** (International Federation for Information Processing); tiene un acuerdo de colaboración con **ACM** (Association for Computing Machinery), así como acuerdos de vinculación o colaboración con **AdaSpain**, **AIZ**, **ASTIC** e **HispanLinux**, junto a la que participa en **ProInnova**.

Consejo Editorial
 Antoni Carbonell Nogueras, Juan Manuel Cueva Lovelle, Juan Antonio Esteban Iriarte, Francisco López Crespo, Celestino Martín Alonso, Josep Molas i Bertrán, Olga Pallás Codina, Fernando Píera Gómez (Presidente del Consejo), Ramón Puigjaner Trepal, Miquel Sàrries Grifó, Asunción Yturbe Herranz

Coordinación Editorial
 Llorenç Pagés Casas <pages@ati.es>
Composición y autedición
 Jorge Llácer Gil de Ramates
Traducciones
 Grupo de Lengua e Informática de ATI <<http://www.ati.es/gl/lengua-informatica/>>, Dpto. de Sistemas Informáticos - Escuela Superior Politécnica - Universidad Europea de Madrid
Administración
 Tomás Brunete, María José Fernández, Enric Camarero, Felicidad López

Secciones Técnicas - Coordinadores
Acceso y recuperación de la información
 José María Gómez Hidalgo (Universidad Europea de Madrid), <jmgomez@uem.es>
 Manuel J. Peña López (Universidad de Huelva), <manuel.mano@diestia.uhu.es>
Administración Pública electrónica
 Francisco López Crespo (MAE), <flc@ati.es>
 Gumersindo García Arribas (MAP), <gumersindo.garcia@map.es>
Arquitecturas
 Enrique F. Torres Moreno (Universidad de Zaragoza), <enrique.torres@unizar.es>
 Jordi Tubellà Morgadas (DAC-UPC), <jordi@ac.upc.es>
Auditoría IT/ITe
 Marisa Touriño Troilito, <marinatourino@marinatourino.com>
 Manuel Palao García-Suelto (ASIA), <manuel@palao.com>
Berecho e tecnologías
 Isabel Hernando Colliázos (Fac. Derecho de Donostia, UPV), <ihernando@legalek.net>
 Elena Davara Fernández (de Marcos Davara & Davara), <edavara@davara.com>
Escadanza Universitaria de la Informática
 Joaquín Epeleta Mateo (CPS-UZAR), <epeleta@posta.unizar.es>
 Cristóbal Pareja Flores (DSIF-UCM), <cpareja@sip.ucm.es>

Entorno digital personal
 Alonso Alvarez García (TID), <aag@tid.es>
 Diego Gachet Paez (Universidad Europea de Madrid), <gachet@uem.es>
Bastión del Conocimiento
 Joan Baiget Solé (Cap Gemini Ernst & Young), <joan.baiget@ati.es>
Informática y Filosofía
 José Ángel Olivas Varela (Escuela Superior de Informática, UCLM)
 Karim Gherrab Martin (Indra Sistemas)
Informática Gráfica
 Miguel Chover Selles (Universitat Jaume I de Castellón), <chover@lsi.uji.es>
 Roberto Vivó Herrando (Eurographics, sección española), <rvivo@dsic.upv.es>
Ingeniería del Software
 Javier Dolado Cosin (DLSI-UPV), <dolado@si.ehu.es>
 Luis Fernández Sanz (PRIS-El-UEM), <lufern@dpriis.esi.uem.es>
Inteligencia Artificial
 Vicente Boti Navarro, Vicente Julián Inglada (DSIC-UPV)
 <vboti.vinglada@dsic.upv.es>

Interacción Persona-Computador
 Julio Abascal González (F-UPV), <julio@sti.ehu.es>
 Jesús Lorés Vidal (Univ. de Lleida), <jesus@eup.udl.es>
Lengua e Informática
 M. del Carmen Ugarte García (IBM), <cuarte@ati.es>
Lenguajes Informáticos
 Andrés Marín López (Univ. Carlos III), <amarin@it.ucm.es>
 J. Ángel Velázquez Iturbide (ESCET-URJC), <a.velazquez@escet.urjc.es>
Lingüística computacional
 Xavier Gómez Guinovart (Univ. de Vigo), <xgdc@uvigo.es>
 Manuel Palomar (Univ. de Alicante), <mpalomar@disi.ua.es>

Mundo estudiantil
 Adolfo Vázquez Rodríguez (Rama de Estudiantes del IEEE-UCM), <a.vazquez@ieeee.org>
 Federico G. Mon Troiti (RIT-SI) <gnu.fede@gmail.com>
Profesión Informática
 Rafael Fernández Calvo (ATI), <rfcalvo@ati.es>
 Miquel Sàrries Grifó (Ayto. de Barcelona), <msarries@ati.es>
Redes y servicios telemáticos
 José Luis Marzo Lázaro (Univ. de Girona), <joseluis.marzo@udg.es>
 Javier Solé Pareta (DAC-UPC), <pareta@ac.upc.es>
Seguridad
 Javier Arellito Bertolin (Univ. de Deusto), <jarellito@eside.deusto.es>
 Javier López Muñoz (ETSI Informática-UMA), <jlm@lcc.uma.es>

Sistemas de Tiempo Real
 Alejandro Alonso Muñoz, Juan Antonio de la Puente Alfaro (DIT-UPM), <jalonso.jpunte@dit.upm.es>
Software Libre
 Jesús M. González Barahona, Pedro de las Heras Quirós (GSYC-URJC), <jmgonz@gsyc.es>
Tecnología de Objetos
 Jesús García Molina (DIS-UM), <jmolina@um.es>
 Gustavo Rossi (LIFA-UNLP, Argentina), <gustavo@sol.info.unlp.edu.ar>
Tecnologías para la Educación
 Juan Manuel Dodero Beardo (UC3M), <dodero@inf.uc3m.es>
 Julia Minguillón i Alfonso (UOC), <jminguillona@uoc.edu>
Tecnologías y Empresa
 Didac López Butifull (Universitat de Girona), <didac.lopez@ati.es>
TIC y Turismo
 Andrés Aguayo Maldonado, Antonio Guevara Plaza (Univ. de Málaga) <aguayo.guevara@lcc.uma.es>

Las opiniones expresadas por los autores son responsabilidad exclusiva de los mismos. **Novática** permite la reproducción, sin ánimo de lucro, de todos los artículos, a menos que lo impida la modalidad de © o copyright elegida por el autor, debiéndose en todo caso citar su procedencia y enviar a **Novática** un ejemplar de la publicación.

Coordinación Editorial, Redacción Central y Redacción ATI Madrid
 Padilla 66, 3º dcha., 28006 Madrid
 Tfn. 91 4029391; fax. 91 3093685 <novatica@ati.es>
Composición, Edición y Redacción ATI Valencia
 Av. del Reino de Valencia 23, 46005 Valencia
 Tfn./fax 963303092 <secreval@ati.es>
Administración y Redacción ATI Cataluña
 Ciudad de Granada 131, 08018 Barcelona
 Tfn. 93 4125235; fax 93 4127713 <secregan@ati.es>
Redacción ATI Andalucía
 Isaac Newton, s/n, Ed. Sadleir,
 Isla Cortija 41092 Sevilla, Tfn./fax 954 460779 <secreand@ati.es>
Redacción ATI Aragón
 Lagasca 9, 3-B, 50006 Zaragoza.
 Tfn./fax 976235181 <secreara@ati.es>
Redacción ATI Asturias-Cantabria <gp-astucant@ati.es>
Redacción ATI Castilla-La Mancha <gp-clmancha@ati.es>
Suscripción y Ventas
 <<http://www.ati.es/novatica/interes.html>>, o en ATI Cataluña o ATI Madrid
Publicidad
 Padilla 66, 3º dcha., 28006 Madrid
 Tfn. 91 4029391; fax. 91 3093685 <novatica.publicidad@ati.es>
Imprenta
 Berria S.A., Juan de Austria 66, 08005 Barcelona.
Distribución legal: B 15.154-1975 -- ISSN: 0211-2124; CODEN NOVACV
Partidas: "Pioneros TIC" (Charles Babbage), RFCalvo / (C) Rafael Fernández Calvo 2006
Diseño: Fernando Agresta / © ATI 2006

editorial

El tema del canon digital no cesa en resumen > 02

La Web que cambió nuestras vidas > 02

Llorenç Pagés Casas

noticias IFIP

Creación del TC14: Comité Técnico en Computación aplicada al entretenimiento > 03

Pedro Antonio González Calero

monografía

Servicios Web

(En colaboración con UPGRADE)

Editores invitados: José Carlos del Arco, Jesús Arias Fisteus, Óscar Corcho, Jorge Cardoso

Presentación. Los Servicios Web desde un punto de vista empresarial y académico > 04

José Carlos del Arco, Jesús Arias Fisteus, Óscar Corcho, Jorge Cardoso

Servicios Web: introducción y estado del arte > 06

Óscar Corcho García, José Carlos del Arco Prieto, Jesús Arias Fisteus

El W3C y la estandarización en los Servicios Web > 11

Martín Álvarez Espinar

Experiencias de integración e interoperabilidad en el Entorno Sanitario > 15

Antonio García Landeira

Desarrollo dirigido por modelos de propiedades extra-funcionales para Servicios Web: un caso de estudio desde las perspectivas del servicio y del cliente > 19

Guadalupe Ortíz Bellot, Juan Hernández Núñez

Hacia un proceso automatizado de trading > 26

Pablo Fernandez Montes, Manuel Resinas Arias de Reyna, Rafael Corchuelo Gil

Servicios Web Semánticos con WSMO > 30

Holger Lausen, Jos de Bruijn, Uwe Keller, Rubén Lara

Hacia la selección semántica de servicios para integración B2B > 34

Andreas Friesen, Kioumars Namiri

Aprovechando modelos de e-commerce para el desarrollo de aplicaciones basadas en Servicios Web > 40

Abraham Nieva de la Hidalga, Liping Zhao, Pedro R. Falcone Sampaio

secciones técnicas

Ingeniería del Software

Obtención de requisitos en ingeniería Web > 45

Stephanos Mavromoustakos, Katerina Papanikolaou

Tecnologías para la Enseñanza

¿Pueden los ordenadores evaluar automáticamente preguntas abiertas? > 50

Diana Pérez Marín, Enrique Alfonso Cubero, Pilar Rodríguez Marín

Tecnologías y Empresa

Impacto de la implantación distribuida de Sistemas Integrados de Gestión > 54

Salvador Bueno Ávila, José Luis Salmerón Silvera

Referencias autorizadas > 58

sociedad de la información

Futuros emprendedores

RingCode2Job: manejando una impresora multifuncional mediante interfaz web > 64

Mikel Fernández de Matauco Amestoy, Irantzu Marquinez Nieto

Novática interactiva

Blogs: en la vanguardia de la nueva generación web > 68

Antonio Fumero, Fernando Sáez Vacas

Los Servicios Web en la práctica > 74

Foro de Debate

Programar es crear

Cuadrados (CUPCAM 2005, problema fase local UCM, solución) > 75

Luis Hernández Yáñez, Cristóbal Pareja Flores

asuntos interiores

Coordinación editorial / Fe de erratas / Socios Institucionales > 77

Mikel Fernández de Matauco Amestoy, Irantzu Marquín Nieto

Facultad de Ingeniería – ESIDE, Universidad de Deusto, Bilbao

{matauco, imarquinez}@gmail.com

1. Introducción

El mercado actual de impresoras multifuncionales (*Multifunctional Printers*, MFP) ofrece un amplio espectro de productos, dotados cada día de más funcionalidad. Es así como nos encontramos con MFPs cada vez más capaces, ofreciendo al usuario la posibilidad de fotocopiar, imprimir, escanear, enviar fax, encuadernar, grapar, etc.

Sin embargo, el aumento del número de servicios ofrecidos va asociado a un incremento de complejidad cada vez más evidente. La dificultad que entraña el manejo de las MFPs se ve reflejada en que no todo el mundo es capaz de salir airoso ante la necesidad de utilizarlas. Este hecho se constata en que cada vez es más frecuente la figura del encargado de manejarla. Nos encontramos aquí ante un empleado que destina parte de su tiempo productivo a la realización de fotocopias, impresión de documentos, envío de faxes, etc. Esto se traduce en pérdida de dinero y recursos para la empresa.

En el sistema RingCode2Job el usuario de un equipo multifuncional encuentra todas las funciones, opciones de trabajo y posibilidades de finalizado en una intuitiva y sencilla interfaz Web. De esta manera el usuario crea su "job" mediante la definición de múltiples tareas, tantas como desee. Las tareas pueden aplicarse página por página (por ejemplo, "imprimir" copias de las páginas 1 a 5; escanear a carpeta la 2 y la 3; enviar por mail la 5; imprimir 2 veces la 6 y enviarla por fax"). De esta manera, se simplifica considerablemente el manejo de las impresoras multifuncionales, ahora cualquiera puede hacer uso de ellas y conseguir sacar el máximo rendimiento.

Con este sistema la empresa propietaria de la impresora multifuncional no deberá destinar ningún recurso para el manejo de la misma, es decir, la interacción usuario/MFP queda reducida a su mínima expresión.

2. Arquitectura del sistema

En la **figura 1** se puede observar la arquitectura de RingCode2Job, en ella se aprecian claramente los dos elementos que componen el sistema (la MFP y el Servidor). Vemos también que varios usuarios pueden estar haciendo uso simultáneo del sistema a través de sus navegadores (*Web browsers*).

RingCode2Job: manejando una impresora multifuncional mediante interfaz web

Resumen: Actualmente, las impresoras multifuncionales (MFPs) han dejado de ser simples impresoras, o fotocopiadoras, pasando a convertirse en un compañero más de la oficina. Esta integración de las MFPs en el trabajo del día a día, se ha visto favorecida por el considerable aumento de su funcionalidad. No obstante, este incremento de capacidad, ha supuesto que la interacción con estas máquinas, se haya vuelto más compleja. El sistema RingCode2Job potencia la funcionalidad de las MFPs, y facilita su configuración y manejo, mediante la definición de tareas en una intuitiva interfaz web, minimizando la interacción Usuario-MFP.

Palabras clave: Jakarta Struts, MFP, Servicios Web, TRIP, Xlet.

Autores

Mikel Fernández de Matauco Amestoy realizó sus estudios de 5º curso de Ingeniería Informática en la Facultad de Ingeniería ESIDE de la Universidad de Deusto en Bilbao durante el curso escolar 2005-2006. Posteriormente, presentó su proyecto fin de carrera junto a Irantzu Marquín Nieto (puesto que en la Universidad de Deusto los proyectos fin de carrera pueden ser presentados por parejas) basado en sus trabajos alrededor del diseño e implementación de RingCode2Job. Trabaja actualmente para la empresa Ibermática en un proyecto para el Gobierno Vasco, empleando las mismas tecnologías utilizadas previamente en la realización de RingCode2Job, lo que según él ha facilitado enormemente su incorporación a este proyecto.

Irantzu Marquín Nieto realizó sus estudios de 5º curso de Ingeniería Informática en la Facultad de Ingeniería ESIDE de la Universidad de Deusto en Bilbao durante el curso escolar 2005-2006. Posteriormente, presentó su proyecto fin de carrera junto a Mikel Fernández de Matauco Amestoy (puesto que en la Universidad de Deusto los proyectos fin de carrera pueden ser presentados por parejas) basado en sus trabajos alrededor del diseño e implementación de RingCode2Job. Trabaja actualmente para la empresa Ibermática en un proyecto para el Gobierno Vasco, empleando las mismas tecnologías utilizadas previamente en la realización de RingCode2Job, lo que según ella ha facilitado enormemente su incorporación a este proyecto.

Los usuarios haciendo uso de cualquier Web browser definen trabajos cuya configuración se envía al servidor mediante HTTP Post, que como respuesta devuelve un RingCode en formato PDF. El usuario imprimirá este Ringcode y lo añadirá al legajo de originales sobre los que desee aplicar dicho trabajo. A este usuario o cualquier persona encargada solo le quedará colocar en el *alimentador automático* (ADF) de la impresora multifuncional la pila de legajos. Tras pulsar el botón 'Start' se iniciará el procesamiento de estos documentos. La MFP mediante HTTP Post envía las imágenes a interpretar al servidor y este tras haberlas interpretado devuelve un XML con la información correspondiente a las tareas que la MFP debe realizar. Posteriormente la impresora aplica a los documentos el trabajo que se le haya ordenado. Una vez concluido el trabajo, es decir, al detectar un nuevo RingCode, o al finalizarse los folios de la bandeja de entrada, la impresora da la orden de notificar al creador del trabajo la finalización del mismo mediante el envío de un email.

2.1. Diseño de la parte servidora

Para la realización de la parte servidora se ha empleado el *framework* Struts, una implementación eficiente de la arquitectura de Modelo 2 siguiendo el patrón MVC, que nos permite crear aplicaciones Web altamente modularizadas y perfectamente separadas en capas (ver **figura 2**).

Struts se compone, en principio, de un servlet controlador, los beans necesarios para pasar información entre el modelo y la vista, y una serie de ficheros de configuración. El servlet de Struts actúa como controlador común para toda la aplicación. Es el encargado de recibir las peticiones de los clientes, invocar a los objetos adecuados para realizar el procesamiento (la lógica), disponer los resultados en forma de beans y dirigir estos a la página JSP (*Java Server Page*) correspondiente para que, haciendo uso de ellos, genere la presentación a devolver (la vista).

Para unir todos estos aspectos, Struts hace uso de una serie de ficheros XML de confi-

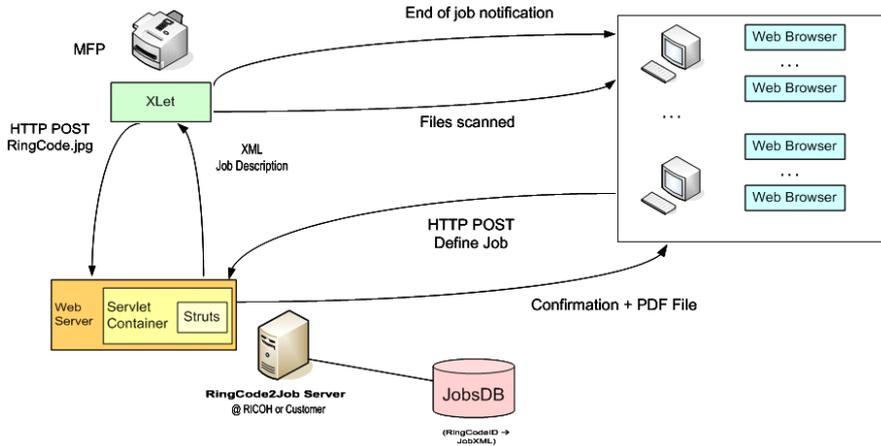


Figura 1. Esquema de la arquitectura de RingCode2Job

guración en los que se definen como han de realizarse todos los pasos anteriormente mencionados, de manera que, por ejemplo, el modificar posteriormente qué página JSP queremos que se encargue de generar la presentación, es cuestión tan sólo de cambiar un par de líneas.

El corazón de Struts es su fichero de configuración, el fichero struts-config.XML. Este fichero, que ha de situarse en la carpeta WEB-INF de la aplicación Web, es leído por Struts en la carga de la aplicación, lo cual implica que los cambios realizados en él no

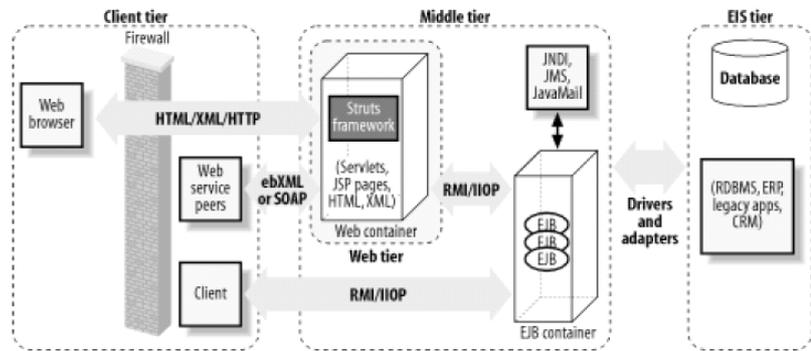


Figura 2. Marco de trabajo de Struts utilizado dentro del nivel Web

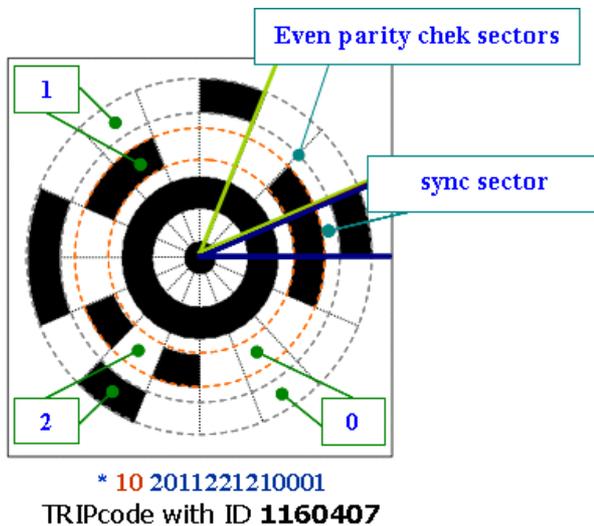


Figura 3. Esquema de un RingCode

serán efectivos a menos que reiniciemos el servidor de JSPs. Además, se apoya también en el fichero Web.XML situado en la misma carpeta, como veremos más adelante.

Una aplicación Struts se basa en una serie de componentes que sirven para implementar el llamado Modelo 2, que es en realidad una arquitectura basada en el conocido patrón MVC (*Modelo-Vista-Controlador*). Estos

componentes son los siguientes:

1. Un servlet controlador.
2. Un conjunto de clases Action.
3. Un conjunto de clases ActionForm.
4. Un conjunto de JSPs.
5. Un archivo de configuración que relaciona todos estos elementos.

Básicamente, Struts funciona de la siguiente manera: el servlet controlador, cuya definición se halla en el archivo Web.XML, escucha todas las peticiones realizadas a la aplicación y de entre todas las peticiones recibidas, aquellas que cumplan un cierto criterio son atendidas en función de las reglas del archivo de configuración (struts-config.XML), mientras que las demás se ejecutan de manera normal. Una petición

será evaluada contra struts-config.XML haciendo uso de una arquitectura propia del Modelo 2, como se expone a continuación:

1. Se buscará en el archivo de configuración una regla cuyo nombre coincida con la página solicitada.
2. Se creará el ActionForm correspondiente, que es el Javabean que usaremos para pasar información entre la clase Action y el JSP.
3. Se creará la clase Action correspondiente, que será la encargada de implementar la lógica de proceso. Una vez finalizado el procesamiento, devolverá los resultados en el Javabean anteriormente creado.
4. Se pasará el control a la página JSP indicada en la regla, la cual recogerá el Javabean y realizará la visualización de su contenido.

De esta manera estamos consiguiendo separar totalmente la vista del procesamiento, que es

	Xlet	main() application
Ciclo de Vida	Loaded, Paused, Active y Destroyed	main y exit
Num. de aplicaciones en VM	Múltiples aplicaciones Xlet	Básicamente una aplicación por VM

Tabla 1. Comparación entre una aplicación Java y un Xlet

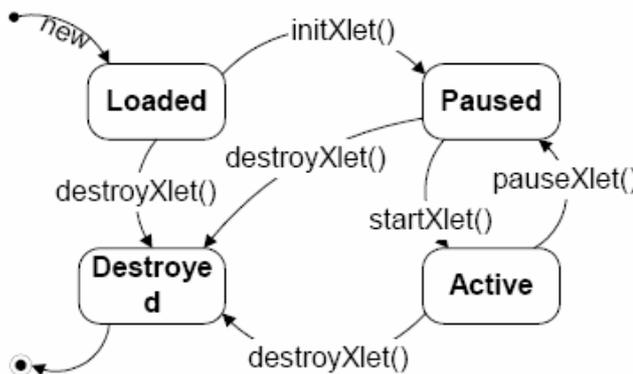


Figura 4. Diagrama de transición entre estados.

precisamente el objetivo del Modelo 2. La generación de los códigos de barras que devuelve la parte servidora al usuario en formato PDF, se crean mediante la utilización del Sistema TRIP. TRIP es un Sistema Open Source destinado a la identificación y localización de objetos señalizados mediante códigos de barras circulares 2D o RingCodes.

Estos códigos circulares (ver figura 3) que representan un número ternario, están compuestos de dos anillos de codificación de 16 bits, que se componen de:

- 1 sector de sincronización.
- 2 sectores de paridad.

De esta manera se consiguen 3^{13} , es decir 1.594.323 posibles códigos. TRIP utiliza conocidos algoritmos de procesamiento de imágenes como *Thresholding*, *edge detection* y *ellipse fitting*.

En la parte servidora de RingCode2Job se hace uso del sistema TRIP tanto para generar los RingCodes correspondientes a los trabajos definidos por el usuario, como para el reconocimiento e interpretación de los mismos cada vez que la MFP así lo requiera. Esta interpretación consiste en extraer el valor de la ID que representa dicho RingCode y devolver como respuesta el XML asociado al *job* con la misma ID de la tabla *jobs* de la base de datos.

2.2. Diseño de la parte cliente

La MFP es considerada la parte cliente de RingCode2Job a la MFP. Gracias al módulo Java del que disponen, las nuevas MFPs pueden albergar aplicaciones denominadas Xlets.

Un Xlet es una aplicación en la que queda claramente definido un ciclo de vida (ver tabla 1 y figura 4) que es controlado por el XletManager (Controlador de Ciclo de Vida). Los diferentes estados por los que puede pasar un Xlet son: Loaded, Paused, Active y Destroyed.

- Estado Inicializado (Loaded): se genera

una instancia de la aplicación Xlet, pero no se inicializa.

- Estado Pausado (Paused): el Xlet es inicializado y pausado con el mínimo uso de recursos.

- Estado Activo (Active): el Xlet es ejecutado normalmente proporcionando su servicio.

- Estado Destruído (Destroyed): el Xlet se destruye así mismo y libera sus recursos. Este estado sólo se alcanza una vez.

En la figura 5 mostramos el diagrama de clases interno de un Xlet, mientras que en la figura 6 podemos ver en un diagrama de secuencia su ciclo de vida y los estados por los que transita. Por último, el proceso que siguen los Xlets usados en el proyecto se especifica en el diagrama de flujo de la figura 7.

3. El Concurso

El sistema RingCode2Job lo creamos para presentarlo a la segunda edición del concurso de programación "Powered by U!" (2006) que organizan Ricoh y Sun Microsystems, en el que resultó ganador nacional, y por tanto finalista europeo.

Como estudiantes de 5º de Ingeniería Informática, ha sido mucho lo que hemos aprendido durante la elaboración de este sistema, pero consideramos tan importantes los conocimientos adquiridos, como la experien-

cia vivida al participar en este concurso. La realización del presente proyecto nos ha ayudado a dar un paso más en nuestra formación como Ingenieros Informáticos y como personas, ya que nos ha servido para conocer más de cerca el mundo empresarial al que en breve nos íbamos a incorporar.

En definitiva ha sido un trabajo duro que ha merecido la pena, y por ello animamos a los estudiantes de informática que estén en sus últimos años de carrera a presentarse a este concurso o a otros similares, ya que el esfuerzo realizado, al menos en nuestro caso, se ha visto recompensado con creces. Queremos agradecer a Ricoh, a Sun Microsystems y a todos aquellos que han hecho posible este concurso la oportunidad que nos han brindado.

4. Conclusiones

Han sido muchas cosas las que nos han entusiasmado en la realización de este sistema pero sin duda la que más nos ha enriquecido como desarrolladores ha sido el descubrimiento de Struts como *framework* de desarrollo de aplicaciones Web. Nos parece sorprendente como este *framework* antes desconocido para nosotros, simplifica enormemente la tarea del desarrollador, proporcionándole una herramienta sencilla y eficiente, para realizar tareas que hasta ahora han sido tediosas y engorrosas como son la creación y gestión de formularios Web.

Tan sorprendente como esta herramienta, ha sido descubrir el mundo de las impresoras multifuncionales, objetos estos a los que sólo les creíamos capaces de escanear, fotocopiar, ..., es decir trabajos elementales, que se les presuponen. Pero la máquina Java que pueden incorporar, característica desconocida para la mayor parte de usuarios de las mismas, las dota de un potencial inmenso, ofreciendo a los Ingenieros Informáticos un nuevo campo de trabajo y a los usuarios ampliar de manera considerable la funcionalidad de sus MFPs. Para finalizar decir que RingCode2Job abre un nuevo mundo de

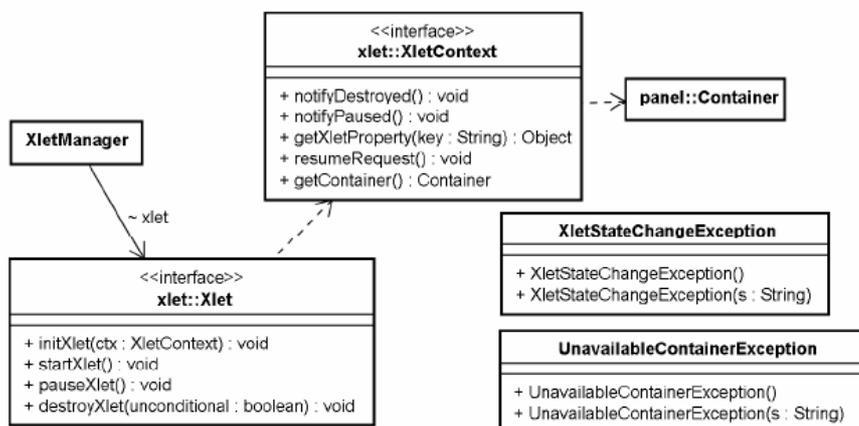


Figura 5. Diagrama de clases de un Xlet.

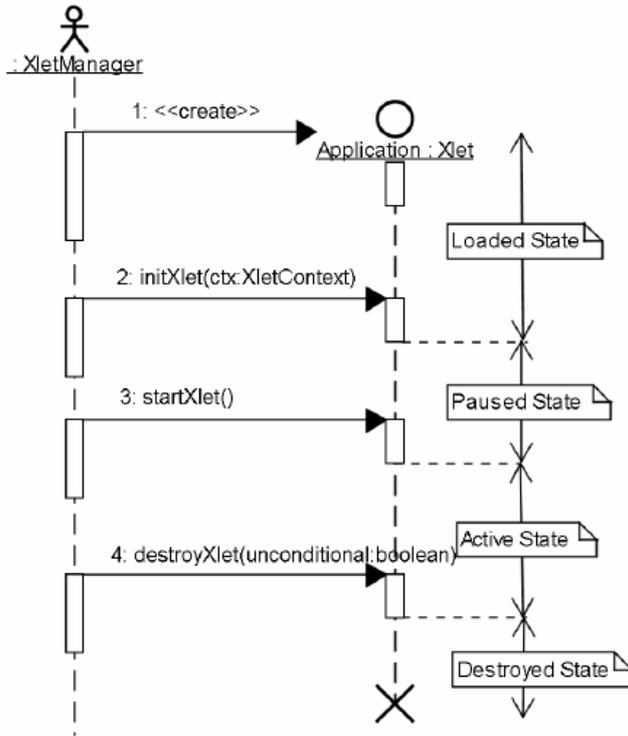


Figura 6. Diagrama de secuencia de un Xlet.

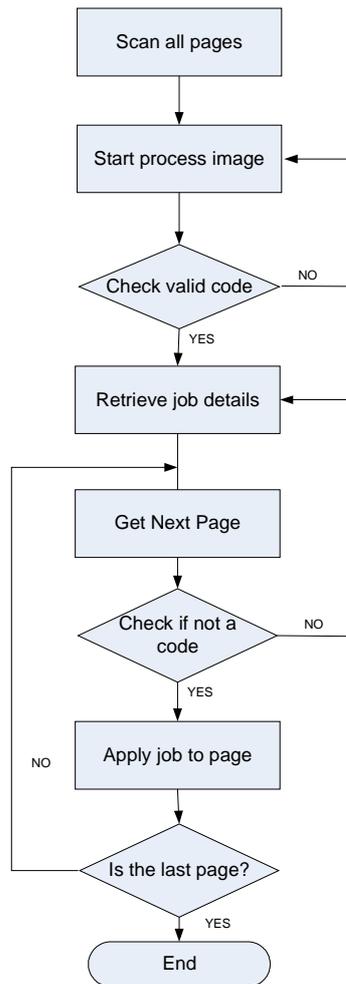


Figura 7. Diagrama de flujo que representa la funcionalidad del Xlet

facilidades para el uso de las MFPs, mediante el paradigma "Show RingCode & Clic", convirtiendo los equipos multifuncionales en máquinas inteligentes que hacen que el día a día de una oficina sea más agradable.

Glosario

ADF: alimentador automático de documentos, es decir, bandeja por la que la MFP recoge el papel.

Bean: componente software que tiene la particularidad de ser reutilizable y así evitar la tediosa tarea de programar los distintos componentes uno a uno.

Java Server Pages (JSP): es la tecnología para generar páginas web de forma dinámica en el servidor, basada en scripts que utilizan una variante del lenguaje Java.

Modelo2: se denomina así a la variante de construir aplicaciones usando a la vez servlets y JSPs siguiendo el patrón MVC.

MFP: nombre que reciben en la actualidad las impresoras multifuncionales, que son aquellas cuya función no se limita a la de impresión, sino que permiten además la ejecución de otras tareas como envío de fax, escaneado, fotocopiado...

MVC: Modelo-Vista-Controlador. Es un patrón de diseño de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos de forma que las modificaciones al componente de la vista pueden ser hechas con un mínimo impacto en el componente del modelo de datos.

Servlet: objeto que corre dentro del contexto de un servidor de aplicaciones y extiende su funcionalidad.

TRIP: sistema Open Source destinado a la identificación y localización de objetos señalizados mediante códigos de barras circulares 2D.

Xlet: nombre que reciben las aplicaciones Java que se ejecutan en el interior de las MFPs.

Referencias

Chuch Cavaness. *Jakarta Struts.* Anaya Multimedia, 2005.

TRIP: a Vision Sensor System. <<http://paginas personales.deusto.es/dipina/tripweb/>>.

Ricoh JavaTM Student Contest, Concurso de Estudiantes <<http://edu.ricoh-developer.com/index.html>>.

Apache Tomcat <<http://tomcat.apache.org/>>.

Apache Struts <<http://struts.apache.org/>>.