

Capítulo 5.

Implementación de HuSystem

Como ya se mencionó en el capítulo anterior, el sistema se dividió en dos componentes, cuya implementación será comentada en este capítulo, para revisar partes del proceso de codificación y de su uso.

Para implementar el sistema se eligió el lenguaje de programación Java, pues ofrecía múltiples beneficios a las dos partes del sistema. El sistema de reconocimiento de caracteres empleado en las máquinas esclavas es OmniPage Pro 7.0 de Caere Corporation.

5.1 HuProcessor.

Para el HuProcessor, la ventaja de usar Java era su sencillo manejo del paralelismo, pues cuenta con una clase llamada hilo (thread) cuyos atributos los puede heredar casi cualquier clase creada por el programador.

Para la implementación del sistema se usaron diversas herramientas de desarrollo, se codificó y diseñó la interfaz de usuario empleando el ambiente integrado de desarrollo CodeWarrior, provisto por Metrowerks Inc., y el ambiente de programación visual de Symantec, Visual Café, además del compilador de Java desarrollado por Apple: el MRJ SDK 2.0. Todos ellos cumpliendo con las especificaciones del software usado para su desarrollo: el JDK 1.1.3 de Sun.

Tanto el HuProcessor como el HuSeeker fueron compilados además en el compilador del JDK de Sun en máquinas Solaris provistas por el laboratorio.

Las pruebas de ejecución se efectuaron tanto en Apple como en Solaris, obteniendo tiempos de ejecución muy superiores en las máquinas producidas por Sun Microsystems.

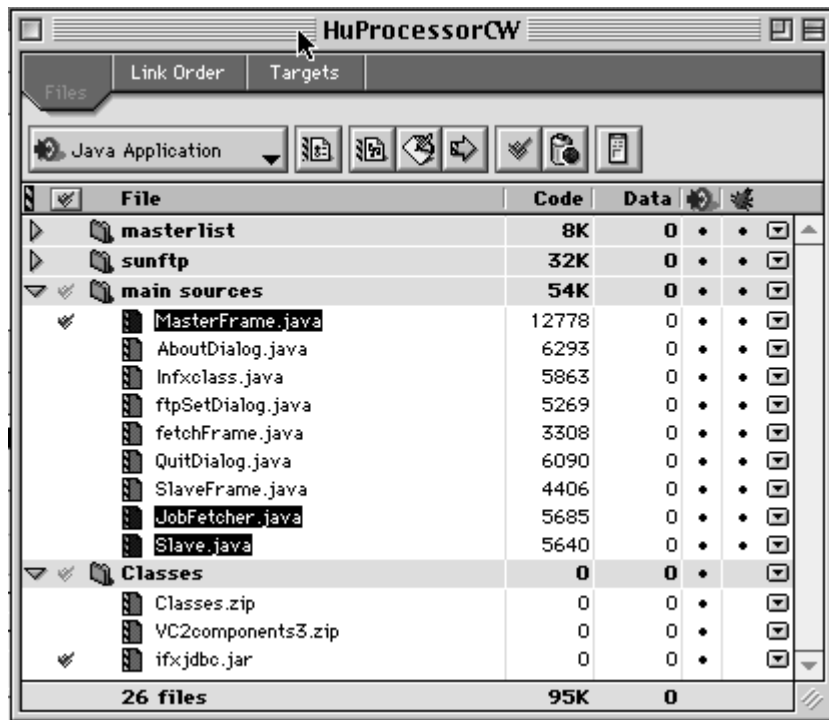


Figura 5.1: Las clases modeladas para el HuProcessor se presentan en la implementación (señaladas con el fondo inverso), en la ventana del ambiente de desarrollo.

Podemos ver en la Figura 5.1 el paso transparente del diseño hacia la implementación, pues las clases que en el diseño del modelo OMT se plantearon como Maestro, Esclavo y Recolector (ver Figura 4.5), ahora corresponden a MasterFrame, Slave y JobFetcher.

El HuProcessor es un sistema orientado para usuarios expertos, por ello su interfaz puede parecer no muy amigable, aunque sí es muy informativa.

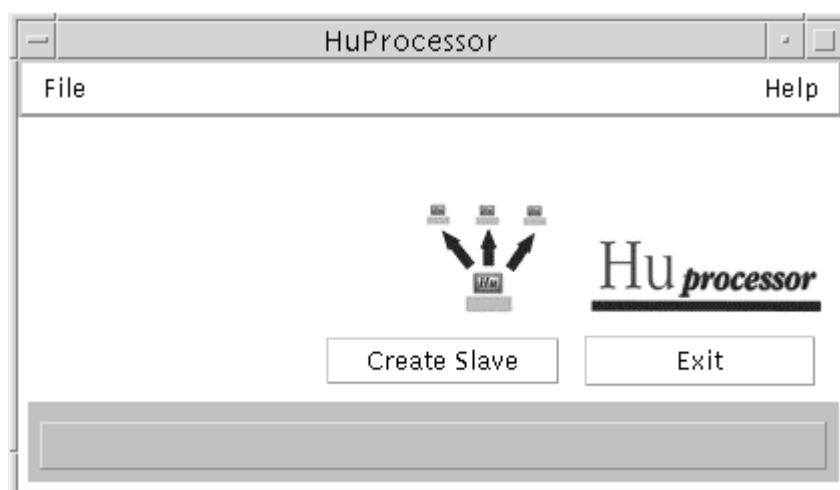


Figura 5.2 Ventana Principal del HuProcessor

La ventana principal (Figura 5.2) ofrece sólo las opciones indispensables para operar el sistema, por medio del menú colgante se pueden establecer los parámetros necesarios para la operación del sistema, y los dos botones que contiene dan idea de la sencillez del manejo del sistema.

En esta ventana se pueden crear esclavos para enviar, monitorear y recoger resultados de máquinas esclavas con reconocedores de texto. Es necesario anteriormente ajustar los parámetros para que todos los esclavos sepan dónde encontrar las imágenes a reconocer y las imágenes a desplegar.

Las imágenes a reconocer son imágenes TIFF a una resolución de 300 puntos por pulgada, con un tamaño promedio de 300 bytes. Las imágenes GIF son las que verá el usuario, pues su tamaño es menor, cada imagen en formato TIFF tiene una imagen igual en formato GIF.

En el momento en el que se presiona el botón llamado "Create Slave", la ventana de la Figura 5.3 aparece para introducir los datos de conexión con el esclavo, lo que se hace por medio del protocolo FTP [Postel & Reynolds 1985].

A screenshot of a dialog box titled "Where is the slave?". It contains four text input fields: "FTP Server:" with the value "140.148.225.6", "Login:" with the value "is092756", "Password:" with the value "*****", and "Default dir:" with the value "ir". At the bottom of the dialog are two buttons: "Cancel" and "OK".

Figura 5.3: Ventana de ajuste para la creación de un esclavo.

Para manipular archivos con el protocolo FTP se utilizó el conjunto de clases no documentadas llamadas Sun.net.FTP, que están incluidas dentro de todas las máquinas virtuales de Java. Estas clases tienen métodos muy básicos para las operaciones FTP, por ello, se tuvieron que adicionar diversas funciones para permitir intercambios de archivos entre varias máquinas, interpretación de listados de archivos para distintas plataformas. Fue decepcionante saber que de las plataformas evaluadas, pocos servidores cumplían con todas las especificaciones mínimas del protocolo FTP.

En Macintosh se probaron cuatro servidores de FTP: WebStar, Net Presentz, El servidor de Telnet de NCSA, y el "plug-in" para FTP del Personal Web Server de Microsoft.

WebStar de StarNine Corporation fue el más compatible, teniendo aún así algunas deficiencias en su implementación del protocolo. Su debilidad es no contar con soporte para cambiar a directorios superiores usando el comando "cd..", de tal manera en que todas las operaciones con directorios se deben hacer tomando en cuenta la ruta de acceso (filepath) absoluta que asigna el servidor al conectarse.

Tabla 5.1 Resultados de las pruebas de servidores FTP.

	WebStar	Net Presenz	NCSA Telnet	Microsoft PWS	FTP Sun
Se pudo conectar	si	si	si	si	si
Se pudo listar archivos	si	si	no	no	si
Se pudo mandar archivos	si	si	no	no	si
Se pudo obtener archivos	si (lento)	si (lento)	no	si	si
Velocidad satisfactoria.	si	si	no	no	si

La Tabla 5.1 describe las cualidades y deficiencias de los diferentes servidores FTP probados. A algunos de ellos no se pudo acceder usando las clases de FTP de Java, debido, tal vez a errores a nivel bajo de la comunicación por sockets.

Todos los servidores FTP descritos anteriormente son de plataforma Mac, excepto el FTP de Sun. No se hicieron pruebas en máquinas Windows porque no se tenía disponible una PC en red.

El HuProcessor usa para las transferencias de archivos el protocolo FTP, pues es el más aceptado por su amplio uso en la red. Así obtiene las imágenes de un servidor remoto, y las manda a las máquinas esclavas. Al concluir la máquina esclava con su trabajo, se recolectan los resultados, y se insertan a la base de datos usando JDBC.

JDBC es una Interfaz para Programación de Aplicaciones (API), que sirve como interfaz para acceder a bases de datos por medio de SQL. JDBC ha sido desarrollado por Sun y sus socios de Java. Este API provee a los programadores una interfaz uniforme hacia un amplio rango de bases de datos relacionales, y provee una base común sobre la que se pueden crear interfaces y herramientas de alto nivel que usen bases de datos transparentemente, JDBC está incluido dentro de la distribución del JDK 1.1 [Sun Microsystems 1998b].

JDBC sólo es un estándar, por sí sólo no puede conectarse a cualquier base de datos, necesita librerías para entender el protocolo de comunicación de cada Manejador de Base de Datos, en este caso se usaron librerías para Informix Universal Server.

El laboratorio cuenta con el manejador de base de datos Informix Universal Server, el cual incluye una API para comunicarse con el manejador usando Java por medio de JDBC. Sin embargo, esta implementación es lenta, además de incómoda, pues requiere que un proceso extra se ejecute en la máquina servidor, que sirve como intermediario entre las aplicaciones escritas por el desarrollador y el manejador de base de datos.

Durante el curso de la implementación, Informix [Informix Corporation 1998] hizo disponible una nueva interfaz JDBC para comunicarse con Informix Universal Server, la cual no necesita de intermediarios (middleware), y se puede usar en cualquier plataforma de hardware que soporte Java. Esta fue la interfaz que se usó tanto en el HuProcessor como en el HuSeeker.

El desarrollo del HuProcessor fue evolucionando también en la etapa de la implementación. Entre las cosas que se tuvieron que contemplar fue que probablemente por alguna falla ajena al sistema, éste dejara de funcionar. Para solucionar este problema, se optó por crear una bitácora que registrara los trabajos concluídos y los trabajos incompletos, de tal manera que si la máquina maestra llegara a fallar, el sistema podría recuperarse y continuar trabajando desde el punto en el que se quedó anteriormente.

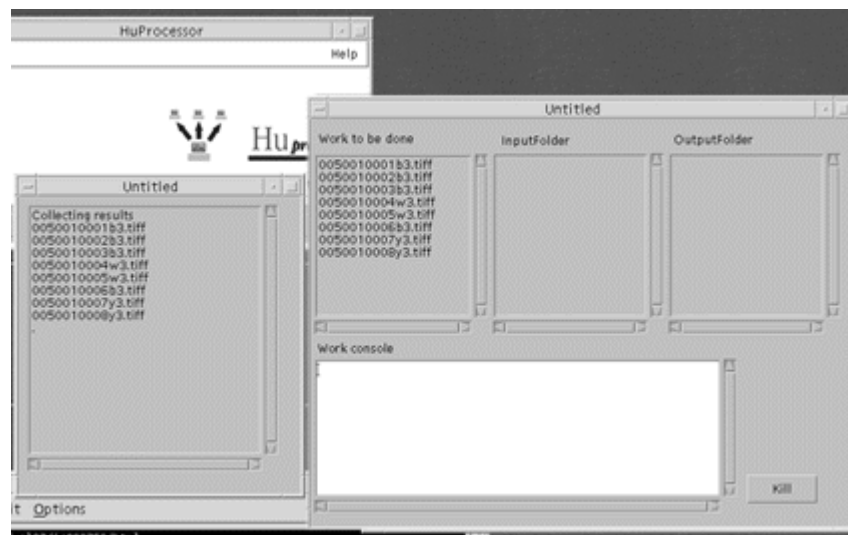


Figura 5.4: Ventanas del Recolector y esclavo, de derecha a izquierda.

El monitoreo del trabajo que hace la máquina esclava se hace por medio de una ventana llamada Esclavo, que contiene cuatro áreas de texto, las cuales sirven para observar la lista de trabajo actual del esclavo, así como los archivos enviados a la máquina esclava y los

archivos de resultados que produce. La cuarta área es una consola, que imprime los mensajes básicos que indican los pasos que se ejecutan en este proceso.

Podemos también observar en la Figura 5.4 (a la izquierda) la ventana del Recolector, que aparece para recoger los resultados del lote actual de archivos en proceso.

5.2 HuSeeker

El HuSeeker fue diseñado para ejecutarse en Internet, contemplando como usuarios a gente interesada en botánica y botánicos autorizados para modificar el sistema miembros de la biblioteca digital florística.

Para el HuSeeker, la ventaja de usar el lenguaje Java es que puede ser ejecutado casi por cualquier usuario en la red, usando un navegador como Netscape Communicator o Microsoft Internet Explorer.

Es importante mencionar que el applet de HuSeeker no puede ser ejecutado en todos los browsers existentes actualmente. Esto se debe a que a pesar de que el lema de Java hacia los desarrolladores es: "Codifique una vez, ejecute en cualquier lugar" (write once, run anywhere), la portabilidad de este lenguaje se ve disminuída por las diferentes implementaciones que diversas compañías tienen del mismo.

En el caso de Microsoft, esta compañía no integra aún soporte para la ejecución de *Métodos de Invocación Remota* (RMI).

No obstante esto, Microsoft Internet Explorer es el único Browser que provee soporte para la ejecución de applets con el modelo de eventos del JDK 1.1.x (Java Development Kit) propuesto por Sun [Brewer, 1998].

El applet del HuSeeker, según Netscape Communications Corporation puede ser ejecutado en todas las versiones de Netscape Communicator 4.5 para las diferentes plataformas, excepto Apple Macintosh y Windows 3.1.

En estos vacíos entra el navegador de Microsoft, Internet Explorer 4.01, el cual, de acuerdo a las pruebas ejecutadas en este trabajo, ejecutó sin problemas el applet.

Se hicieron pruebas también en el browser desarrollado por Apple llamado Cyberdog, el cual, mediante tecnología OpenDoc puede ejecutar applets con la máquina virtual del sistema operativo; sin embargo, este applet falló.

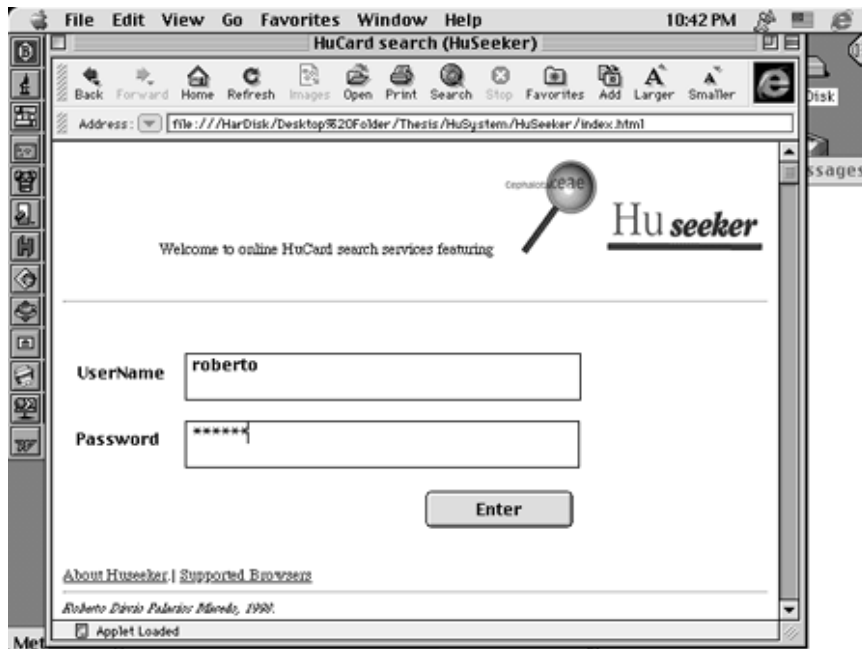


Figura 5.5: Página principal de acceso al sistema HuSeeker. Aquí, el buscador no trabaja en complemento de los sistemas HuBrowser y 3DTree.

En la Figura 5.5 se puede observar la pantalla inicial del sistema, que pide la identificación del usuario, para saber si es un editor o un usuario común. El objetivo a largo plazo es perfeccionar el texto asociado a la tarjeta. Esta tarea se asigna a los usuarios expertos que se identifican en esta pantalla inicial, quienes estarán autorizados para modificar el texto. Esto tendrá efecto para mejorar los resultados en las búsquedas de tarjetas.

La Figura 5.6 muestra la ventana de búsqueda del sistema, la cual permite hacer consultas por fracciones de texto contenido en la tarjeta, posteriormente llena el componente de selección con los identificadores de las tarjetas que cumplen al contener en ellas el texto que el usuario busca.

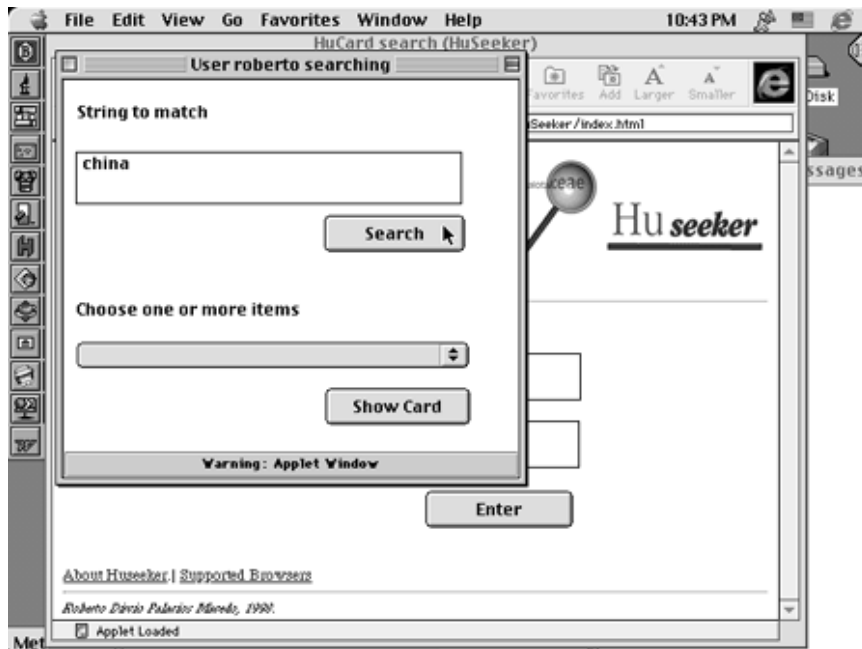


Figura 5.6: La ventana de búsqueda de HuSeeker.

El usuario, al haber efectuado una búsqueda y seleccionado alguna tarjeta puede ver el contenido de la tarjeta encontrada, y si es un usuario experto tiene la opción de modificarla usando el botón que aparece en la parte inferior.(Ver Figura 5.7)

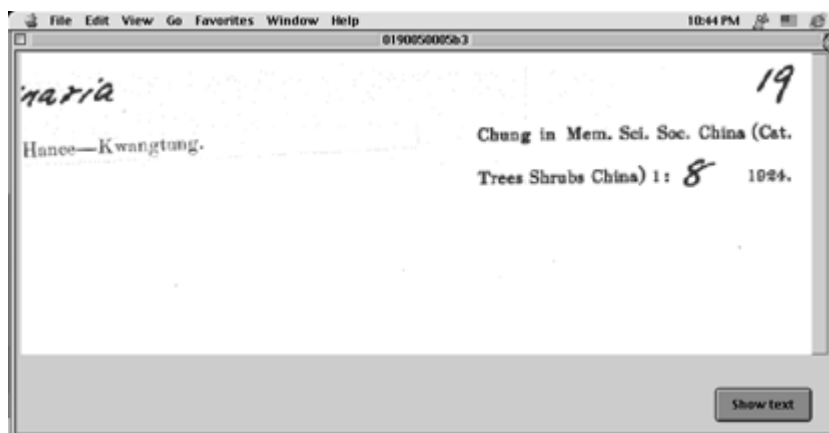


Figura 5.7: despliegue de la tarjeta de Hu elegida.

Si el usuario decide hacer cambios al texto, presionará el boton para mostrar el texto, el cual le presentará una ventana en la que aparecerá el texto editable.

En caso de que el texto haya sido editado por otro usuario, se mostrará el nombre del último usuario que modificó el texto. De esa manera se controlará la autoría de la edición en la colección de tarjetas.

Dircio Palacios Macedo, R. 1998. **Reconocimiento y Consulta de Imágenes Textuales en Bibliotecas Digitales**. Tesis Licenciatura. Ingeniería en Sistemas Computacionales. Departamento de Ingeniería en Sistemas Computacionales, Escuela de Ingeniería, Universidad de las Américas-Puebla. Diciembre.

Derechos Reservados © 1998, Universidad de las Américas-Puebla.