

Universidad Autónoma Metropolitana

Unidad Azcapotzalco

División de Ciencias Básicas e Ingenierías
Licenciatura en Ingeniería en Computación

Proyecto Terminal

Aplicación móvil para la sincronización de archivos

Rodrigo Ramírez Reyes

Matrícula: 204308981

Dra. Beatriz Adriana González Beltrán

Departamento de Sistemas

No. económico: 30246

México, D.F. a 21 de Abril del 2010

Tabla de contenidos

Resumen	4
Introducción	5
Objetivos	6
Desarrollo del proyecto	6
Desarrollo de la aplicación cliente	6
Diagrama de casos de uso	6
Diagrama de funcionamiento	15
Diagrama de arquitectura	16
Diagrama de navegación	17
Diagrama de clases	17
Desarrollo del servidor de sincronización de archivos.....	20
Descripción técnica del proyecto	20
XML	21
Seguridad (SHA-1)	21
MIDP	22
Midlet.....	22
JAR y JAD	22
Conclusiones	23
Bibliografía	24

Índice de figuras

Figura 1: Diagrama de casos de uso del sistema.....	7
Figura 2: Pantalla del menú “Ver archivos”	8
Figura 3: Pantalla del explorador de archivos del móvil.....	9
Figura 4: Pantalla del explorador de archivos del servidor	10
Figura 5: Opciones de sincronización.....	11
Figura 6: Opciones de reemplazo	12
Figura 7: Opciones de Sincronización en modo Descarga.....	14
Figura 8: Opciones de Sincronización en modo Carga	15
Figura 9: Diagrama de funcionamiento de la aplicación	16
Figura 10: Diagrama de la arquitectura del sistema	17
Figura 11: Diagrama de navegación del MIDlet.....	18
Figura 12: Diagrama de clases en formato UML del sistema	19

Resumen

La aplicación móvil para la sincronización de archivos permite sincronizar archivos entre un dispositivo móvil y una computadora que tiene un servicio web que ofrece una interfaz para la sincronización de archivos.

La aplicación móvil cuenta con la capacidad de explorar los archivos tanto del servidor como del dispositivo móvil en el que se ha instalado. Las funciones principales de la aplicación son descargar archivos desde el servidor y cargar archivos hacia el servidor de sincronización de archivos, así como la sincronización la cual involucra las dos opciones anteriores.

La aplicación para la sincronización de archivos tiene la cualidad de poder seleccionar la dirección de red donde se encuentra el servidor de sincronización de archivos, ya sea de forma local o por medio de la web, con la posibilidad de usar cualquier forma de conectividad inalámbrica soportada por el dispositivo móvil.

La aplicación ha sido desarrollada en Java (J2ME), siendo compatible con la mayoría de los dispositivos móviles actuales.

Palabras clave

Sincronización de archivos, aplicación móvil, servicios web, MIDP, XML, J2ME

Introducción

El número de usuarios de teléfonos móviles crece día a día, así como el desarrollo de aplicaciones móviles, incluyendo las aplicaciones web. Este crecimiento ha sido posible gracias al acceso a redes inalámbricas tales como WiFi, 3G y HSDPA.

La ventaja de utilizar las aplicaciones móviles es que se pueden realizar actividades sencillas desde cualquier lugar, con la restricción de que su funcionamiento y desempeño está limitado por el rendimiento del procesador del móvil y la memoria RAM. Sin embargo, el hardware de los dispositivos portátiles está aumentando muy rápidamente, por lo que existen aplicaciones móviles con las cuales el usuario ya es capaz de ingresar a redes sociales, portales bancarios, realizar trámites, revisar el correo electrónico, enviar archivos adjuntos, ver vídeo, entre otras actividades.

En este trabajo se explora la sincronización de archivos de manera que se puedan tener en múltiples dispositivos de almacenamiento una misma versión de la información que se ha dispuesto a sincronizar.

Entre las aplicaciones de sincronización de archivos que existen actualmente son Dropbox[1], Windows Live Sync de Microsoft[2], Synctoy[3], Synclicity[4], Windows Live Folder Share[5], y SyncBack[6], las cuales son herramientas de sincronización para computadoras. Estas aplicaciones permiten la sincronización entre un directorio y un espacio en un servidor externo, sin embargo tienen la desventaja de que el servidor es predeterminado y existe un límite de almacenamiento por parte del servidor.

Existen otras aplicaciones para la sincronización de información que están dirigidas principalmente a tareas de sincronización del correo electrónico y de los contactos. Sin embargo, tienen la desventaja de que se sincronizan mediante una conexión por cable USB o vía Bluetooth. Algunos ejemplos de aplicaciones móviles que realizan la sincronización de archivos son: EyeOS[7], Microsoft SQL Server 2005 Compact Edition Access Database Synchronizer[8] y CalSync60[9], entre otras. En Internet existe diversa información acerca de los servicios de sincronización de información utilizando servicios Web desde aplicaciones móviles pero solamente entre un dispositivo móvil y una computadora.

La aplicación desarrollada en este proyecto no depende de un solo servidor previamente configurado por el desarrollador de la aplicación sino que permite que el usuario defina el servidor al que desea conectarse, siempre y cuando el servidor ofrezca los servicios necesarios para la sincronización de datos para realizar la conexión desde un dispositivo móvil.

Objetivos

El objetivo general de este proyecto fue desarrollar una aplicación móvil que permite sincronizar archivos mediante servicios web por medio de las redes inalámbricas.

El proyecto se dividió en tres funciones generales que se listan a continuación:

- Establecer la conexión al servidor.
- Gestionar las bitácoras entre el cliente y servidor.
- Realizar la transferencia de archivos entre el cliente y el servidor.

Desarrollo del proyecto

Este proyecto involucra dos aplicaciones, la aplicación móvil de sincronización de archivos que reside en la parte del dispositivo portátil (cliente), y la aplicación de sincronización de archivos que reside en una computadora (servidor).

Las características de la aplicación de sincronización de archivos son las siguientes:

- Interoperabilidad. La aplicación funciona con ayuda de los servicios Web utilizando archivos XML permitiendo la operación con múltiples plataformas.
- Multiplataforma. Al ser desarrollada en Java, lo que permite funcionar donde se encuentre instalada la máquina virtual de java.
- Seguridad. La aplicación envía cifrada la información con el uso del algoritmo SHA-1.
- Configurable. Servidor definido por el usuario permitiendo la opción de ingresar a un servidor web que ofrezca la interfaz de sincronización de archivos.

A continuación se describe la aplicación por parte del cliente y, posteriormente, el lector encontrará una breve descripción de la aplicación por parte del servidor.

Desarrollo de la aplicación cliente

En esta sección se describe la aplicación por parte del cliente.

Diagrama de casos de uso

En particular, el diagrama de casos de uso [10] que se muestra a continuación permite visualizar las acciones principales que se realizan en la aplicación móvil, visualizando de forma gráfica la interacción entre los actores y los casos de uso.

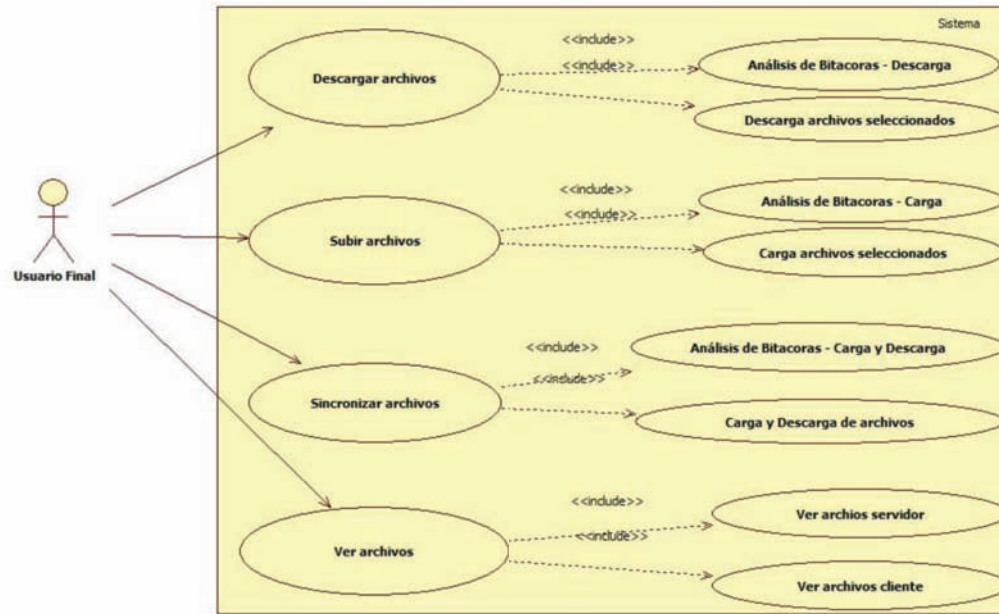


Figura 1: Diagrama de casos de uso del sistema

Los casos de uso vistos en el diagrama anterior se convierten en las acciones principales del sistema. En las siguientes secciones se explica cada caso de uso.

Ver archivos

En la opción de “Ver archivos” se visualizan los archivos del móvil y del servidor (Ver figura 2).



Figura 2: Pantalla del menú "Ver archivos"

Las opciones principales en el menú "Ver archivos" son las siguientes:

- **Ver archivos del móvil:** Explorador del directorio de archivos del móvil.
- **Ver archivos del servidor:** Explorador del directorio de archivos del servidor.

Visualizar archivos del móvil

En la opción "Ver archivos del móvil" se pueden ver los archivos del móvil (Ver Figura 3).

Proyecto terminal: Aplicación móvil para la sincronización de archivos



Figura 3: Pantalla del explorador de archivos del móvil

Ver archivos del servidor

En la opción "Ver archivos del servidor", se visualizan los archivos del servidor (Ver Figura 4).

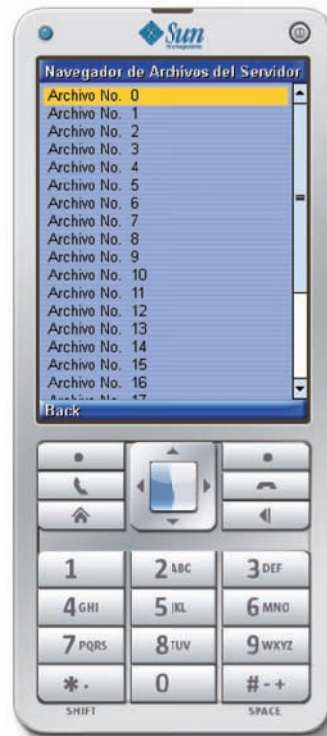


Figura 4: Pantalla del explorador de archivos del servidor

Sincronizar

La opción de sincronizar dentro del menú principal permite realizar tres acciones principales las cuales se muestran en la siguiente imagen (Ver figura 5), siendo “Sincronización sin reemplazo”, “Sincronización con reemplazo” y “Sincronización con reemplazo y preguntar en caso de reemplazo”.

En esta opción, la sincronización se realiza en ambas direcciones, tomando en cuenta la opción seleccionada.



Figura 5: Opciones de sincronización

Las tres opciones dentro del menú Sincronización son las siguientes:

Sincronización sin reemplazo

La opción de sincronización sin reemplazo permite realizar la transferencia de archivos entre el cliente y el servidor en ambas direcciones, transfiriendo los archivos que solo se encuentran en el cliente o solo en el servidor.

Sincronización con reemplazo

La opción de sincronización con reemplazo permite realizar la transferencia de archivos entre el cliente y el servidor en ambas direcciones de los archivos, que además de transferir los archivos que solo se encuentran en el cliente o solo en el servidor, se reemplazan los archivos existentes, previa selección, por parte del usuario de la dirección en que serán reemplazados los archivos (Ver Figura 6). Las dos opciones posibles son:

- Reemplazo desde el cliente: En la opción “Reemplazo desde el Cliente”, se reemplazan todos los archivos que ya se encuentren en el servidor con las versiones del cliente.
- Reemplazo desde el servidor: En la opción “Reemplazo desde el Servidor”, se reemplazan todos los archivos que ya se encuentren en el cliente con las versiones del servidor.



Figura 6: Opciones de reemplazo

Al finalizar de realizar dichas acciones, la aplicación mostrará finalmente al usuario si se ha realizado con éxito o no.

Sincronización con reemplazo y preguntar en caso de reemplazo

La opción de “sincronización con reemplazo y preguntar en caso de reemplazo” permite realizar la transferencia de archivos entre el cliente y el servidor en ambas direcciones de los archivos que ambos no tienen, pero reemplazando los archivos que ya tiene el cliente del servidor o viceversa. En esta opción, a diferencia de la “sincronización con reemplazo”, la aplicación preguntará al usuario, por cada archivo que ya exista en ambos directorios, en cual dirección desea realizar el reemplazo, ya sea en dirección del “servidor” (reemplazo desde el servidor hacia el cliente) o del “cliente” (reemplazo desde el cliente hacia el servidor).

Descargar archivos

En la opción “Descargar Archivos” se realiza una comparación entre el directorio de archivos del directorio local y el directorio del servidor de archivos, descargando los archivos que no se encontraban en el directorio del móvil. En esta opción se despliega la ventana de la Figura 7, donde se permite elegir entre tres posibles opciones de sincronización:

- **Sincronización sin reemplazo:** Descarga, desde el servidor, todos los archivos que no se encuentran en el cliente.
- **Sincronización con reemplazo:** Descarga, desde el servidor, los archivos que no se encuentran en el móvil y los archivos que ya se encuentran los reemplaza.
- **Sincronización con reemplazo y preguntar en caso de reemplazo:** Descarga, desde el servidor, los archivos que no se encuentran en el móvil y los archivos que ya se encuentran, la aplicación preguntará al usuario si quiere reemplazar la versión del servidor por la del cliente.



Figura 7: Opciones de Sincronización en modo Descarga

Subir archivos

En la opción "Subir Archivos" se realiza una comparación entre el directorio de archivos local y el directorio del servidor de archivos, subiendo los archivos que no se encontraban en el directorio del servidor de archivos. La aplicación permite al usuario elegir entre tres posibles opciones de sincronización (Ver Figura 8):

- **Sincronización sin reemplazo:** Carga, desde el cliente, todos los archivos que no se encuentran en el servidor.
- **Sincronización con reemplazo:** Carga, desde el cliente, los archivos que no se encuentran en el servidor y los archivos que ya se encuentran los reemplaza.
- **Sincronización con reemplazo y preguntar en caso de reemplazo:** Carga, desde el cliente, los archivos que no se encuentran en el servidor. Los archivos repetidos, la aplicación preguntará al usuario si quiere reemplazar la versión del cliente por la del servidor.



Figura 8: Opciones de Sincronización en modo Carga

Diagrama de funcionamiento

La figura 9 muestra el diagrama de funcionamiento que nos permite visualizar la forma y dirección en que se realiza la sincronización de archivos.

Proyecto terminal: Aplicación móvil para la sincronización de archivos

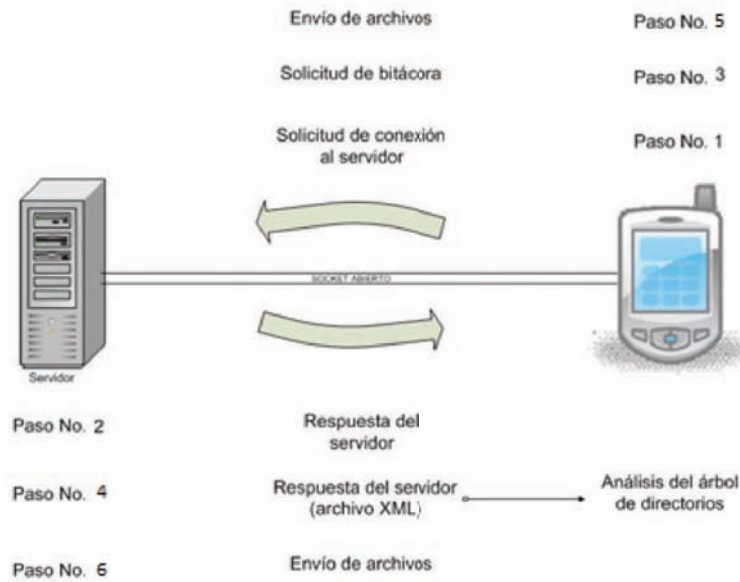


Figura 9: Diagrama de funcionamiento de la aplicación

Diagrama de arquitectura

El diagrama de arquitectura es una vista lógica de los componentes y servicios del entorno en donde se desarrolla y comunica la aplicación con el servidor de archivos (Ver Figura 10). De esta forma podemos visualizar las tecnologías involucradas y la forma de comunicación entre el cliente y el servidor.

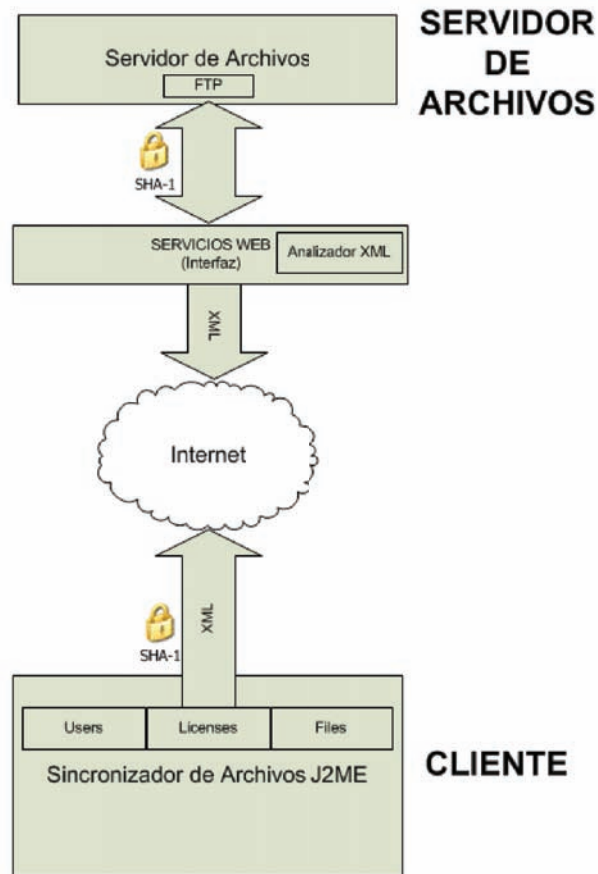


Figura 10: Diagrama de la arquitectura del sistema

Diagrama de navegación

La figura 11 nos muestra el diagrama de navegación de la parte del cliente donde se puede apreciar con mayor claridad el flujo de las actividades de la aplicación móvil, ya sea por decisión del usuario o por procesos de ejecución del sistema.

Para generar el diagrama de navegación se utilizó la herramienta de desarrollo NetBeans definiendo un proyecto MIDP (*Mobile InformationDeviceProfile*)[11]. Ver sección: Justificación MIDP.

Diagrama de clases

La figura 12 muestra el diagrama de clases UML[12] con el objeto de visualizar las relaciones entre las clases de diseño involucradas en la aplicación móvil de sincronización de archivos.

Desarrollo del servidor de sincronización de archivos

El servidor de sincronización de archivos se desarrolló como un servicio Web utilizando *Rest*. La comunicación entre la aplicación cliente y el servidor de archivos se hace a través del envío de solicitudes a través de la Web, utilizando el protocolo http.

Las tareas que realiza el servidor de sincronización de archivos son la comparación de bitácoras, la recepción de archivos por parte del servidor, la descarga de archivos, la validación de usuario, la solicitud del directorio de archivos del servidor, la solicitud de fecha de un archivo, la solicitud de tamaño de un archivo y la verificación de la existencia de un archivo.

Estos servicios web son el intermediario entre el cliente y el servidor de archivos. Se utilizó un servidor ftp para almacenar los archivos de los usuarios.

El servidor FTP[13] utilizado fue la aplicación *Filezilla Server*, donde se requiere inicialmente crear la cuenta del usuario, así como asignar un directorio de archivos dentro del disco duro de la computadora. Este espacio se convierte en su directorio personal por parte del servidor de archivos.

El inicio de sesión se realiza mediante una solicitud a la interfaz del servidor de sincronización de archivos, la cual a su vez realiza un acceso al servidor FTP con la cuenta del usuario, de modo que al iniciar sesión en la aplicación de sincronización se valida que se encuentren disponibles (activos) tanto los servicios web como el servidor.

Descripción técnica del proyecto

El proyecto ha sido desarrollado con una herramienta llamada Netbeans[14] utilizando J2ME[20][22][23] como plataforma de desarrollo.

Se utilizó J2ME debido a que la máquina virtual de Java (JRE) se encuentra disponible en una gran cantidad de dispositivos. J2ME está basado en J2SE con algunas restricciones en cuanto a algunas funciones o declaraciones, por lo que aunque no tenga toda la funcionalidad de J2SE fue suficiente para satisfacer las necesidades de este proyecto.

Para facilitar la programación de la aplicación móvil se utilizó la plataforma de diseño MIDP, que permite generar automáticamente la aplicación (MIDLET).

A continuación se describen, de manera breve, otras tecnologías involucradas en este proyecto.

XML

Una de las ventajas de la aplicación móvil para la sincronización de archivos, es la interoperabilidad, por lo que se eligieron utilizar archivos con formato XML[15][21], para la comunicación entre la aplicación móvil y la interfaz del servidor de archivos.

El formato de XML es un formato universal creado por el W3C. XML permite interactuar entre los distintos lenguajes que existen sin la necesidad de crear adaptaciones entre ellos, siempre y cuando ambos sepan interpretar o definir el formato de XML. Este formato ha sido utilizado para el intercambio de información estructurada, tanto en la Internet como entre aplicaciones o plataformas.

XML fue utilizado en mi proyecto para intercambiar la información generada por los servicios Web. La parte del móvil cuenta con un módulo que permite interpretar la información en el formato XML generando objetos en Java.

Seguridad (SHA-1)

A pesar de que la aplicación es capaz de ser configurada por cualquier servidor que ofrezca los servicios de sincronización necesarios, es requerido que parte de la información sea enviada con algunos métodos de seguridad, como lo es el método SHA-1[16]. Este nivel de seguridad se añadió a la aplicación debido a que la seguridad en J2ME es limitada, de manera muy similar a la que habría en un navegador Web. De manera particular, en este proyecto se ha tomado el módulo de seguridad desarrollado para el entorno de J2SE y se ha adaptado en base al soporte de J2ME para poder añadir los módulos de cifrado de la información, de modo que incluso cuando la información sea leída por un tercero, estará en un formato que solo entiende el cliente y el servidor.

SHA-1 es utilizado en una gran cantidad de sitios web que integran seguridad con SSL, así como en combinación con otros algoritmos de seguridad. Es por eso que se ha implementado un método para la encriptación de datos por medio del algoritmo SHA-1, el cual es irreversible y no es sencillo de que vuelvan a tener la información si es interferida, de modo que solamente tanto el cliente como el servidor son capaces de enviar la información cifrada. SHA-1 cuenta con bloques de 512 bits y emplean 4 funciones primitivas para generar el resumen del mensaje. Procesa alrededor de 80 pasos, por lo que puede ser un 10% más lento que los demás algoritmos de seguridad.

MIDP

El proyecto ha sido desarrollado bajo un esquema llamado MIDP ya que permite trabajar dos entornos de programación, uno directamente sobre el código y otro entorno en forma gráfica. Para el flujo de la aplicación es posible definir un diagrama de flujo para la aplicación. Este entorno cuenta con CLDC (*Connected Limited Device Configuration*) [17].

Un MIDP puede ser manipulado, además en otros ambientes: código, pantalla y analizador. Estos a su vez integran la tecnología de *Java Sun Wireless Toolkit*[18], permitiendo emular la aplicación en un dispositivo, sin tener que transferir la aplicación a un móvil para poder hacer pruebas antes de que sea la versión definitiva. En este proyecto, todas las pruebas se realizaron utilizando este emulador.

La herramienta de desarrollo genera dentro de sus archivos de distribución, los archivos JAR y JAD (Ver justificación JAD y JAR) que son los archivos ejecutables y donde se encuentran todas las clases y recursos, así como los descriptores de estos mismos.

Midlet

Un MIDlet es la aplicación que se genera con un MIDP. Todo MIDlet tiene un ciclo de vida que inicia al desde la ejecución de la aplicación hasta la destrucción de la aplicación regresando el control total al sistema del móvil. Entender el ciclo de vida de un MIDlet es fundamental para su creación. El ciclo de vida define los estados de ejecución de un MIDlet: la creación, el inicio, la pausa, la salida y las transiciones entre estados válidos.

JAR y JAD

La aplicación móvil de sincronización de archivos se encuentra en dos archivos de distribución con extensión jar y jad[20].

El archivo JAD describe un conjunto de MIDlets. Esta descripción incluye el nombre de la suite MIDlet, su ubicación y el tamaño del archivo JAR. También contiene la configuración del perfil del dispositivo para el cual fue desarrollado así como los requisitos del celular. La sintaxis del archivo JAD es similar a la de la clase *java.util.Properties* que encuentran en el entorno de J2SE.

El archivo JAR contiene uno o más MIDlets, especificado en el JAD con el MIDlet-<n> atributo. Un archivo de manifiesto se encuentra en /META-INF/MANIFEST.MF dentro del archivo JAR. El manifiesto tiene la misma sintaxis que el archivo JAD y puede compartir los mismos atributos.

Conclusiones

Las aplicaciones para dispositivos móviles como las PDA y los Smartphone se han vuelto muy populares, que incluso con el gran crecimiento del uso de portátiles (laptops), las aplicaciones móviles se han vuelto igual de necesarias en la vida de cualquier persona, ya que permiten realizar tareas básicas en cualquier lugar siempre y cuando se cuente con conectividad inalámbrica a Internet.

En la actualidad ya se cuentan con varios sistemas operativos para móviles, con los cuáles se están desarrollando aplicaciones que en algunos casos son idénticas a las que se ofrecen para una computadora de escritorio, aunque todavía se han detenido un poco los desarrolladores para realizar aplicaciones más complejas debido al bajo procesamiento de los dispositivos móviles actuales.

Algo que ayuda a las aplicaciones móviles actuales, es la interoperabilidad de los sistemas con el uso de archivos XML que permiten incluso interactuar varias plataformas y lenguajes con un mismo sistema, de modo que al tratarse de archivos de texto estructurado, son fáciles de interpretar, incluso por un dispositivo móvil, de modo que el procesamiento puede ser realizado por parte del servidor y la interpretación de datos por la aplicación cliente.

Bibliografía

- [1] Dropbox. [Online].Disponible: <https://www.dropbox.com/features>
- [2] Windows Live Sync de Microsoft. [Online].Disponible:
http://help.live.com/Help.aspx?mkt=en-us&project=WL_Sync
- [3]Synctoy. [Online].Disponible:
<http://www.microsoft.com/downloads/details.aspx?familyid=c26efa36-98e0-4ee9-a7c5-98d0592d8c52&displaylang=en>
- [4]Synclplicity. [Online].Disponible: <http://manual.synclplicity.com/>
- [5]Windows Live Folder Share. [Online].Disponible:
<http://foldershareteam.spaces.live.com/default.aspx?sa=872141391>
- [6]SyncBack. [Online].Disponible: <http://www.2brightsparks.com/syncback/sbpro-features.html>
- [7]EyeOS. [Online].Disponible: http://eyeos.org/index.php?p=whatiseyeos_feature
- [8]Microsoft SQL Server 2005 Compact Edition Access Database Synchronizer. [Online].Disponible:
<http://www.microsoft.com/downloads/details.aspx?familyid=B967347A-5DD0-445C-8A9F-AEA3DB9EC4BC&displaylang=en>
- [9]CalSync60. [Online].Disponible: <http://www.googasync.com/>
- [10]Diagrama de casos de uso. [Online].Disponible:
<http://www.edicionsupc.es/ftppublic/pdfmostra/OE04304M.pdf>
- [11]Mobile Information Device Profile. [Online].Disponible:
<http://java.sun.com/products/midp/>
- [12]UML. [Online].Disponible: <http://www.uml.org/#UMLProfiles>
- [13]FTP. [Online].Disponible: <http://mina.apache.org/ftpserver/>
- [14]Netbeans. [Online].Disponible: <http://netbeans.org/features/index.html>
- [15]W3C. [Online].Disponible: <http://www.w3.org/standards/xml/core>
- [16]SHA-1. [Online].Disponible:http://www.w3.org/PICS/DSig/SHA1_1_0.html
- [17]Connected Limited Device Configuration. [Online].Disponible:
<http://java.sun.com/products/cldc/>

- [18] Java Sun Wireless Toolkit. [Online]. Disponible:
<http://java.sun.com/products/sjwtoolkit/download.html?feed=JSC>
- [19] Jar/Jad. [Online]. Disponible: <http://java.sun.com/javame/reference/docs/sjwc-2.0-web/docs/ToolsGuide-html/jadtool.html>
- [20] Kim Topley , J2ME in a Nutshell, Ed. O'Reilly Media.
- [21] Abraham Gutiérrez , XML a través de ejemplos, Ed. Alfaomega.
- [22] Michael Kroll , Java 2 Micro Edition (J2ME) Application Development, Ed. SAMS.
- [23] Maximiliano Firtman, Programación para Celulares con Java (J2ME), Ed. MP.

Universidad Autónoma Metropolitana
Unidad Azcapotzalco

División de Ciencias Básicas e Ingenierías
Licenciatura en Ingeniería en Computación

MANUAL DE USUARIO

Aplicación móvil para la sincronización de archivos

Rodrigo Ramírez Reyes
Matrícula: 204308981

Dra. Beatriz Adriana González Beltrán
Departamento de Sistemas
No. económico: 30246

México, D.F. a 21 de Abril del 2010

ÍNDICE

Contenido

INTRODUCCIÓN	3
CONFIGURAR LA CONEXIÓN CON EL SERVIDOR	4
INICIAR SESIÓN.....	6
MENÚ PRINCIPAL	7
VER ARCHIVOS.....	9
VISUALIZAR ARCHIVOS DEL MÓVIL.....	9
VER ARCHIVOS DEL SERVIDOR.....	10
SINCRONIZAR	11
• SINCRONIZACIÓN SIN REEMPLAZO.....	12
• SINCRONIZACIÓN CON REEMPLAZO.....	12
• REEMPLAZO DESDE EL CLIENTE	13
• REEMPLAZO DESDE EL SERVIDOR	13
• SINCRONIZACIÓN CON REEMPLAZO Y PREGUNTAR EN CASO DE REEMPLAZO	13
DESCARGAR ARCHIVOS	14
SUBIR ARCHIVOS	18
SALIR DE LA APLICACIÓN	22
CONSIDERACIONES DEL USO DE LA RED	23

INTRODUCCIÓN

Este manual de usuario describe la aplicación MoFiSync, una aplicación móvil para la sincronización de archivos. De manera general, MoFiSync permite sincronizar archivos entre un directorio local fijo del móvil con un directorio en un servidor de archivos.

De manera particular, MoFiSync ofrece cuatro opciones: “Descargar Archivos”, “Cargar Archivos”, “Ver archivos” y “Sincronizar”.

La opción “Descargar Archivos” realiza una comparación entre los archivos del directorio local y del directorio del servidor, descargando los archivos que no se encontraban en el directorio del móvil.

Por su parte, la opción “Cargar Archivos” realiza una comparación entre los archivos del directorio del servidor y del directorio de archivos local, subiendo los archivos que no se encontraban en el directorio del servidor de archivos.

En cuanto a la opción “Ver archivos” es posible ver el contenido de los directorios del cliente y del servidor con la opción de ver las propiedades de los archivos del cliente.

Por último, la opción “Sincronizar” realiza una comparación los archivos del directorio local y del directorio del servidor de archivos, descargando los archivos que no se encontraban en el directorio del cliente y cargando los archivos que no se encontraban en el directorio del servidor.

A continuación se explica a detalle el uso de MoFiSync.

CONFIGURAR LA CONEXIÓN CON EL SERVIDOR

Para poder configurar la dirección del servidor, deberás presionar el botón de Menú en la pantalla de “Inicio de sesión”, seleccionando la opción “Configurar Servidor” (Figura 1).



Figura 1: Pantalla de Menú de configuración

Una vez que hayas ingresado en la opción de “Configurar Servidor”, te mostrará la siguiente pantalla (Figura 2), donde deberás de ingresar la ruta del servidor de archivos en la etiqueta de texto. Al final de esta sección, podrás ver una serie de ejemplos.

En caso de ser la primera vez que se ingresa a la configuración del móvil, estará en blanco de modo que no podría haber conexión y sería necesario por única vez ingresar la dirección del servidor, sin embargo después de haber configurado, no será necesario volver a hacerlo a menos que el servidor tuviera una dirección de red distinta.



Figura 2: Pantalla de configuración de servidor

En la pantalla de configuración de servidor se mostrará la última configuración del servidor de archivo, de modo que podrá ser editada en caso de querer cambiar el servidor.

Si has ingresado a esta opción, pero no deseas realizar ningún cambio, podrás regresar a la pantalla de inicio de sesión por medio del botón "Cancelar".

Si deseas guardar los cambios en la dirección del servidor, deberás presionar el botón "Menú" y después seleccionar la opción de "Guardar", llevándote a una ventana de confirmación y regresando a la pantalla de inicio de sesión.

Observaciones:

La configuración de la dirección del servidor de archivos se puede modificar, sin embargo el nombre de los servicios web son fijos, por lo que solamente se podrá cambiar la dirección IP o URL del servidor, así como añadir un puerto.

Por ejemplo:

- <http://10.10.10.10>
- <http://192.1.50.50:8080>
- <http://www.archivos.com>
- <http://www.archivos.com:8484>

INICIAR SESIÓN

Al iniciar la Aplicación móvil para la sincronización de archivos, el sistema comienza con una pantalla de inicio de sesión la cual se muestra a continuación (Figura 3).



Figura 3: Pantalla de inicio de sesión

En la pantalla de inicio de sesión se debe ingresar el nombre del usuario y contraseña para acceder a la aplicación. Después presionar el botón de “Menú” y seleccionar “Login” (Ver ejemplo en Figura 4).

Ejemplo de pantalla Login:



Figura 4: Ingreso de datos en la pantalla de inicio de sesión

Esto llevará al usuario a la pantalla del menú principal. (Ver "Menú Principal")

MENÚ PRINCIPAL

En la pantalla de "Menú Principal", se ven las opciones principales del sistema (Ver Figura 5).



Figura 5: Pantalla del menú principal

Una vez iniciada la sesión (Ver "Iniciar Sesión"), se presenta la pantalla del menú principal.

Las opciones principales en el menú principal son las siguientes:

- **Ver archivos:** Explorador del directorio de archivos.
- **Descargar archivos:** Descarga de archivos con la opción de seleccionar los archivos requeridos.
- **Subir archivos:** Carga de archivos con la opción de seleccionar los archivos requeridos.
- **Sincronizar:** Menú de opciones de sincronización tanto en descarga como carga de archivos en una sola operación.
- **Salir:** Finaliza la aplicación.

VER ARCHIVOS

En la opción de “Ver archivos”, podrás visualizar los archivos del móvil y del servidor (Ver figura 6).



Figura 6: Pantalla del menú “Ver archivos”

Las opciones principales en el menú “Ver archivos” son las siguientes:

- **Ver archivos del móvil:** Explorador del directorio de archivos del móvil.
- **Ver archivos del servidor:** Explorador del directorio de archivos del servidor.

VISUALIZAR ARCHIVOS DEL MÓVIL

En la opción “Ver archivos del móvil”, podrás ver los archivos del móvil (Ver Figura 7).



Figura 7: Pantalla del explorador de archivos del móvil

Una vez que se hayan desplegado los archivos, tú puedes seleccionar entre tres opciones:

- **Icono de carpeta con título “..”:** Sube al directorio superior.
- **Icono de archivo:** Muestra las propiedades del archivo.
- **Icono de carpeta con nombre:** Ingresa al contenido del directorio.

VER ARCHIVOS DEL SERVIDOR

En la opción “Ver archivos del servidor”, podrás ver los archivos del servidor (Ver Figura 8).

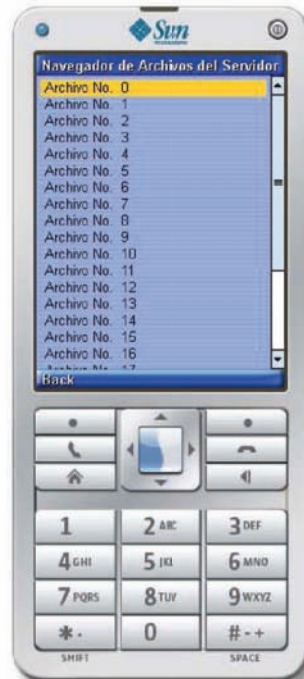


Figura 8: Pantalla del explorador de archivos del servidor

SINCRONIZAR

La opción de sincronizar dentro del menú principal, permitirá realizar tres acciones principales las cuales se muestran en la siguiente imagen (Ver figura 9), siendo "Sincronización sin reemplazo", "Sincronización con reemplazo" y "Sincronización con reemplazo y preguntar en caso de reemplazo".

En esta opción, la sincronización se realiza en ambas direcciones, tomando en cuenta la opción seleccionada.

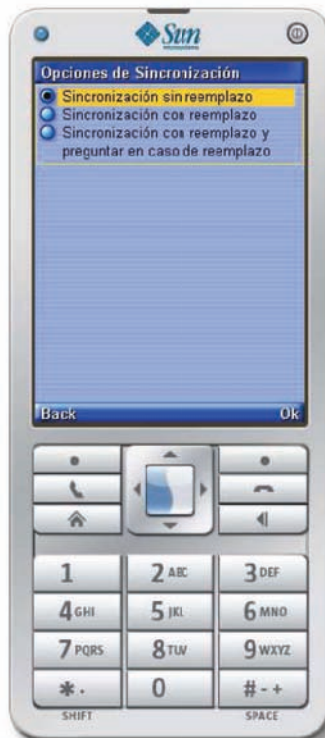


Figura 9: Opciones de sincronización

Las tres opciones dentro del menú Sincronización son las siguientes:

- SINCRONIZACIÓN SIN REEMPLAZO

La opción de sincronización sin reemplazo te permite realizar la transferencia de archivos entre el cliente y el servidor en ambas direcciones de los archivos que ambos no tienen, así como también realiza dichas acciones mostrando finalmente al usuario si se ha realizado con éxito o no.

- SINCRONIZACIÓN CON REEMPLAZO

La opción de sincronización con reemplazo te permite realizar la transferencia de archivos entre el cliente y el servidor en ambas direcciones de los archivos que ambos no tienen, pero reemplazando los archivos que ya tiene el cliente del servidor o viceversa. Es por eso que después de seleccionar esta opción, deberás seleccionar en cual dirección quieres que sean reemplazados los archivos (Ver Figura 10).

Por lo que las dos opciones posibles son:

- REEMPLAZO DESDE EL CLIENTE

En la opción “Reemplazo desde el Cliente”, se reemplazan todos los archivos que ya se encuentren en el servidor con las versiones del cliente.

- REEMPLAZO DESDE EL SERVIDOR

En la opción “Reemplazo desde el Servidor”, se reemplazan todos los archivos que ya se encuentren en el cliente con las versiones del servidor.



Figura 10: Opciones de reemplazo

Al finalizar de realizar dichas acciones, la aplicación mostrará finalmente al usuario si se ha realizado con éxito o no.

- SINCRONIZACIÓN CON REEMPLAZO Y PREGUNTAR EN CASO DE REEMPLAZO

La opción de “sincronización con reemplazo y preguntar en caso de reemplazo” te permite realizar la transferencia de archivos entre el cliente y el servidor en ambas direcciones de los archivos que ambos no tienen, pero reemplazando los archivos que ya tiene el cliente del servidor o viceversa, pero que a diferencia de la “sincronización con reemplazo”, en esta opción, la aplicación te preguntará en cual dirección deseas realizar el reemplazo ya sea en dirección “servidor” (reemplazo desde el servidor hacia el cliente) o “cliente” (reemplazo desde el cliente hacia el servidor) por cada archivo que ya exista en ambos directorios.

DESCARGAR ARCHIVOS

- **Descargar Archivos:** En el modo Descarga, realiza una comparación entre el directorio de archivos del directorio local y el directorio del servidor de archivos, descargando los archivos que no se encontraban en el directorio del móvil.

En la opción de Descargar Archivos se despliegan las siguientes ventanas, donde se permite elegir entre las dos posibles opciones de sincronización (Ver Figura 11):



Figura11: Opciones de Sincronización en modo Descarga

Se ofrecen tres opciones de sincronización en el modo Descarga siendo las siguientes:

- **Sincronización sin reemplazo:** Descarga todos los archivos que no se encuentran en el cliente desde el servidor.
- **Sincronización con reemplazo:** Descargar los archivos que no se encuentran en el móvil y los archivos que ya se encuentran los reemplaza.
- **Sincronización con reemplazo y preguntar en caso de reemplazo:** Descargar los archivos que no se encuentran en el móvil y los archivos que ya se encuentran la aplicación te preguntará si quieres reemplazar la versión del servidor por la del cliente.

Después de haber aceptado las opciones de Sincronización en el modo Descarga, se muestra la siguiente opción (Ver Figura 12).



Figura 12: Aviso de que comenzará la comparación entre directorio local y remoto

En este paso, aparece una ventana que muestra al usuario de que comenzará la conexión con el servidor para poder realizar la comparación entre el espacio de archivos del móvil y del servidor, necesarias para la sincronización.

Si no existe error en la comparación de directorio de archivos local y remoto, la aplicación continúa a la siguiente pantalla mostrada a continuación. En caso de haber error, muestra la notificación de error y regresa al menú principal de la aplicación.

Después, deberás de seleccionar los archivos que deseas descargar, después de haber realizado la comparación de archivos entre ambos directorios, con la opción de seleccionar "Todos", "Ninguno" o seleccionar manualmente los archivos que se quieran descargar. Después deberás de presionar la opción "ok", para continuar (Ver figura 13).



Figura 13: Selección de archivos a sincronizar

En este paso, se lanza una ventana que muestra al usuario de que comenzará la transferencia de archivos en modo Descarga (Ver Figura 14).



Figura 14: Aviso de que la transferencia de datos comenzará en modo Descarga

Si no existe error en la transferencia de archivos, regresa al menú principal de la aplicación. En caso de haber un error en la transferencia, muestra la notificación de error y regresa al menú principal de la aplicación.

Los posibles errores podrían deberse a un error en la apertura de archivos o una falla en la conectividad.

SUBIR ARCHIVOS

- **Subir Archivos:** En el modo Subir Archivos, realiza una comparación entre el directorio de archivos local y el directorio del servidor de archivos, subiendo los archivos que no se encontraban en el directorio del servidor de archivos.

En la opción de Subir Archivos se despliegan las siguientes ventanas, donde se permite elegir entre las dos posibles opciones de sincronización (Ver Figura15).



Figura 15: Opciones de Sincronización en modo Carga

Se ofrecen dos opciones de sincronización en el modo Subir Archivos siendo las siguientes:

- **Sincronización sin reemplazo:** Carga todos los archivos que no se encuentran en el servidor desde el cliente.
- **Sincronización con reemplazo:** Cargar los archivos que no se encuentran en el servidor y los archivos que ya se encuentran los reemplaza.
- **Sincronización con reemplazo y preguntar en caso de reemplazo:** Cargar los archivos que no se encuentran en el servidor y los archivos que ya se encuentran la aplicación te preguntará si quieres reemplazar la versión del cliente por la del servidor.

Después de haber aceptado las opciones de Sincronización en el modo Subir Archivos, se muestra la siguiente opción (Ver Figura 16).

En este paso, aparece una ventana que muestra al usuario de que comenzará la comparación entre los directorios de archivos local y remoto, necesaria para la sincronización.



Figura 16: Aviso de que comenzará la comparación de directorios de archivo local y remota

Si no existe error en la comparación de los directorios de archivos local y remoto, la aplicación continúa a la siguiente pantalla mostrada a continuación. En caso de haber error, muestra la notificación de error y regresa al menú principal de la aplicación.

Los posibles errores de la comparación de directorios de archivos local y remoto, podría deberse a un error en la apertura de archivos, falla en la conectividad o la indisponibilidad del servidor.

Después, deberás de seleccionar los archivos que deseas cargar, después de haber realizado la comparación de archivos entre ambos directorios, con la opción de seleccionar "Todos", "Ninguno" o seleccionar manualmente los archivos que se quieran cargar. Después deberás de presionar la opción "ok", para continuar (Ver Figura 17).



Figura 17: Selección de archivos

En este paso, se lanza una ventana que muestra al usuario de que comenzará la transferencia de archivos en modo Subir Archivos (Ver Figura 18).



Figura 18: Aviso de que la transferencia de datos en modo Subida comenzará

En este paso, se lanza una ventana que muestra al usuario de que comenzará la transferencia de archivos en modo Subir Archivos.

Si no existe error en la transferencia de archivos, regresa al menú principal de la aplicación. En caso de haber un error en la transferencia, muestra la notificación de error y regresa al menú principal de la aplicación.

Los posibles errores podrían deberse a un error en la apertura de archivos o una falla en la conectividad.

SALIR DE LA APLICACIÓN

En la opción "Salir" del menú principal, al seleccionar dicha opción, se muestra la siguiente pantalla (Ver figura 19).

La pantalla que verás es una ventana tipo "acerca de", donde muestra información del autor y la Universidad.



Figura 19: Ventana "Acerca de" previo a finalizar la aplicación

Si seleccionas la opción "ok", se cierra la aplicación, regresando al menú de aplicaciones del sistema operativo del móvil.

CONSIDERACIONES DEL USO DE LA RED

La aplicación móvil requiere del uso de la red, sin embargo la aplicación no restringe al usuario de utilizar cualquiera de los distintos tipos de redes que el móvil soporte (ej. WiFi, 3G, HSDPA, EDGE).

Este tipo de redes podrían generar un costo por la transferencia de datos, a excepción de WiFi, sin embargo se recomienda consultar a su compañía de telefonía móvil para conocer los cargos por uso de la red.

Universidad Autónoma Metropolitana

Unidad Azcapotzalco

División de Ciencias Básicas e Ingenierías
Licenciatura en Ingeniería en Computación

MANUAL DE INSTALACIÓN

Aplicación móvil para la sincronización de archivos

Rodrigo Ramírez Reyes
Matrícula: 204308981

Dra. Beatriz Adriana González Beltrán
Departamento de Sistemas
No. económico: 30246

México, D.F. a 21 de Abril del 2010

Manual de Instalación: MoFiSync

Contenido

INTRODUCCIÓN.....	3
INSTALACIÓN POR CABLE USB	4
EL MÓVIL TIENE UNA APLICACIÓN DE INSTALACIÓN DE ARCHIVOS	4
EL MÓVIL NO TIENE UNA APLICACIÓN DE INSTALACIÓN DE ARCHIVOS	5
INSTALACION POR BLUETOOTH	7
<i>OBSERVACIONES:</i>	8

INTRODUCCIÓN

El manejo de archivos de instalación para móviles, es distinto, de acuerdo a cada sistema operativo, sin embargo la aplicación ha sido desarrollada en Java, siendo compatible con la mayoría de los móviles actuales.

La aplicación es generada en dos archivos , un jar y un jad, siendo requerido en algunos sistemas, solamente el jar, aunque en algunos otros sistemas se requieren de ambos.

El peso aproximado de la aplicación no rebasa los 500kb, por lo que no se requiere tener más de 1MB en espacio en memoria local o en tarjeta extraíble.

En cuanto a las aplicaciones .jar, son fácilmente detectadas por la mayoría de los sistemas para móviles, como archivos de instalación, por lo que explicaré algunos de los pasos para transferir e instalar en los móviles.

Los pasos de instalación mostrados a continuación son genéricos, si desea conocer el nombre exacto de las opciones a seguir, consultar el manual del teléfono donde se quiere instalar.

INSTALACIÓN POR CABLE USB

EL MÓVIL TIENE UNA APLICACIÓN DE INSTALACIÓN DE ARCHIVOS

Para la instalación por medio de cable USB, se deberán de seguir los siguientes pasos dependiendo la situación del móvil:

1. Abrir aplicación para transferencia de archivos de instalación (Ver Figura 1).
2. Conectar el móvil a la PC o laptop.
3. Seleccionar la memoria local o memoria extraíble donde se requiere instalar el archivo.
4. Seleccionar el archivo de instalación en la computadora (Ver Figura 1).

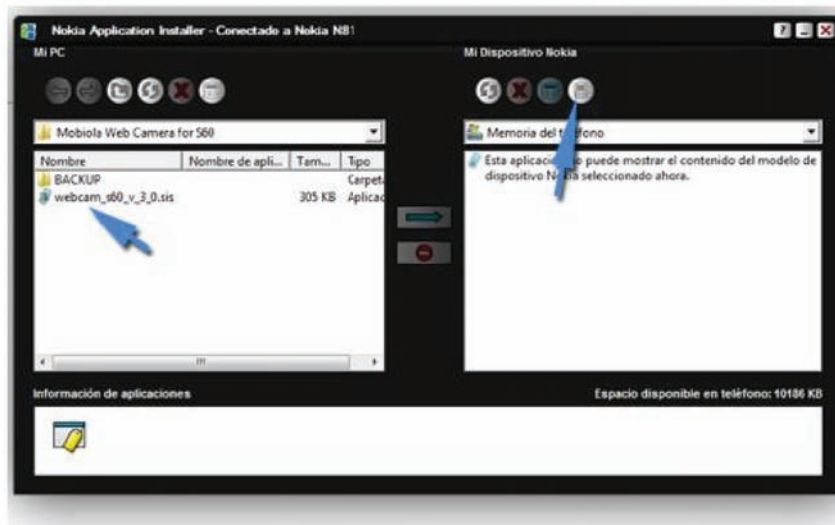


Figura 1: Imagen representativa de instalador de archivos de Nokia

5. Iniciar la transferencia.

Una vez que el archivo se haya transferido, se comenzará la transferencia. Se deberán aceptar los certificados y en tal caso, aceptar si el móvil advierte que la aplicación podrá tener acceso a los directorios del móvil, así como hacer uso de la red.

EL MÓVIL NO TIENE UNA APLICACIÓN DE INSTALACIÓN DE ARCHIVOS

1. Deberás conectar el cable USB a la computadora y al móvil.
2. Después esperar a que la computadora lo detecte como unidad extraíble. En algunos casos deberás de seleccionar esta opción al conectar el móvil con la computadora (Ver Figura 2).

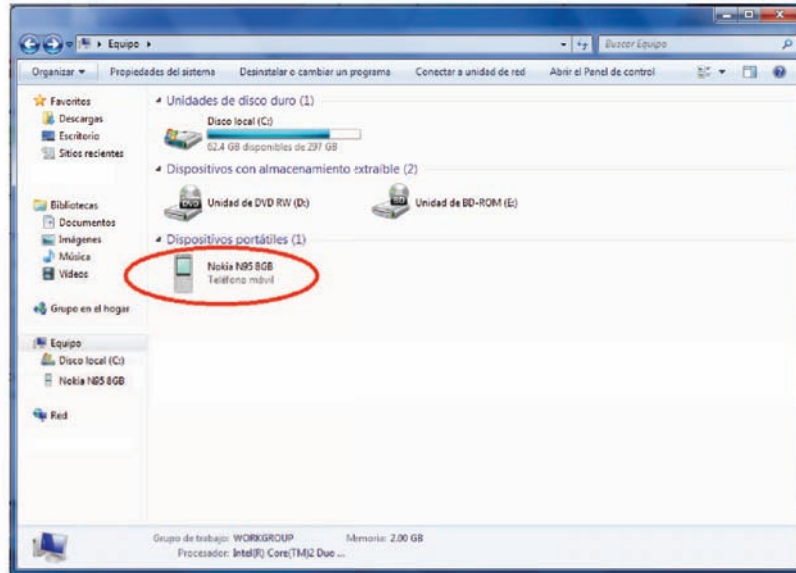


Figura 2: Detección del móvil por medio del explorador del sistema operativo

3. Por medio del explorador de archivos de la computadora, se deberá copiar el archivos jar, jad o ambos, a la memoria local o a la memoria extraíble del móvil (Ver figura 3).

Manual de Instalación: MoFiSync

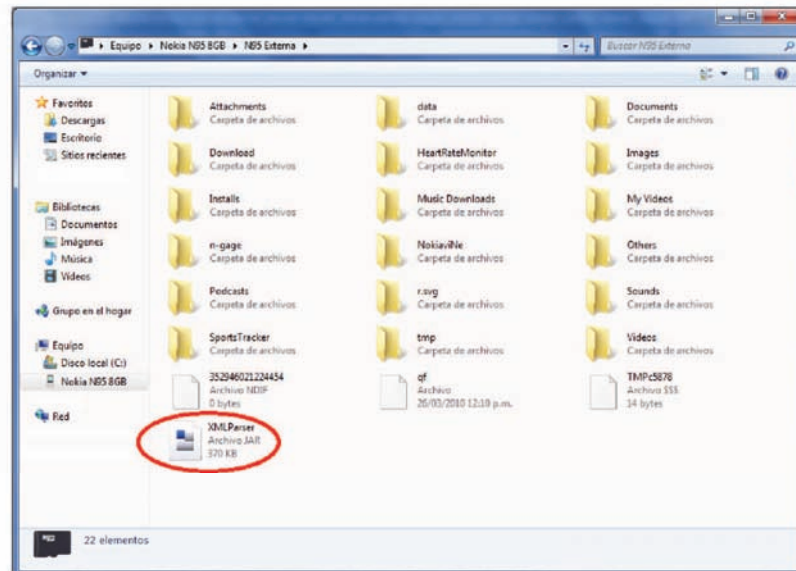


Figura 3: Archivo pegado desde el explorador del sistema operativo

4. Una vez que se haya transferido la aplicación, desde el explorador del móvil, se deberá de ejecutar el instalador, para que el sistema del móvil lo instale.

Una vez que el archivo se haya transferido, se comenzará la transferencia. Se deberán aceptar los certificados y en tal caso, aceptar si el móvil advierte que la aplicación podrá tener acceso a los directorios del móvil, así como hacer uso de la red.

INSTALACION POR BLUETOOTH

Pasos a seguir para instalar por medio de conectividad Bluetooth:

1. Deberás encender el Bluetooth del móvil y asignar un nombre, así como la opción de visibilidad.
2. Se deberá de encender el módulo de Bluetooth en el ordenador.
3. Después, deberás vincular ambos dispositivos, para tener permiso de enviar y recibir archivos (Ver Figura 4).

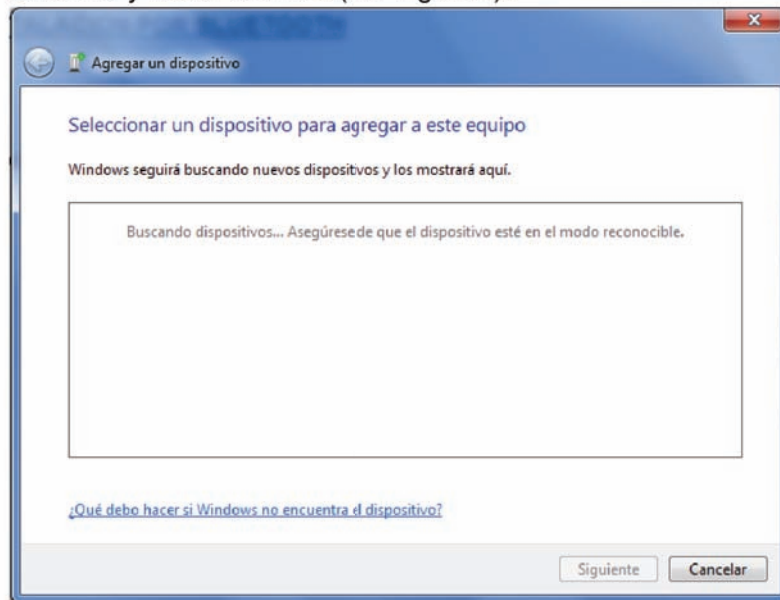


Figura 4: Conectar el ordenador con el dispositivo móvil

4. Una vez que los hayas vinculado satisfactoriamente (Ver Figura 5), deberás seleccionar el archivo y enviar por Bluetooth al móvil.

Manual de Instalación: MoFiSync

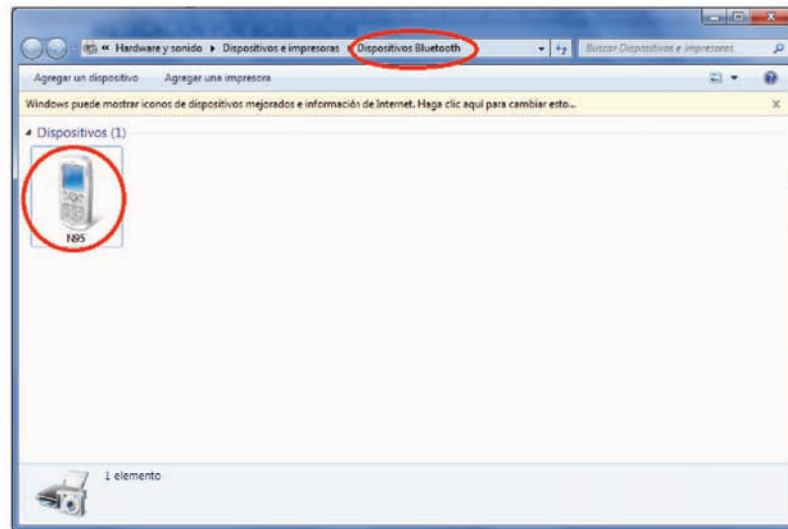


Figura 5: Vinculación entre el móvil y el ordenador por medio de conexión Bluetooth

5. Recibirás la notificación de transferencia en el móvil y deberás de aceptarla.
6. Comenzará la transferencia del archivo.
7. Cuando haya finalizado la transferencia, el móvil te preguntará si deseas instalar la aplicación.
8. Deberás de seleccionar la memoria en que deseas sea instalada.

Se deberán aceptar los certificados y en tal caso, aceptar si el móvil advierte que la aplicación podrá tener acceso a los directorios del móvil, así como hacer uso de la red.

OBSERVACIONES:

Para la instalación por medio de Bluetooth, se deberá de tener una computadora con Bluetooth disponible y un móvil con Bluetooth.

XMLParser
1.0

Generated by Doxygen 1.6.3

Mon Apr 26 17:08:09 2010

Contents

1	Namespace Index	1
1.1	Package List	1
2	Class Index	3
2.1	Class List	3
3	File Index	5
3.1	File List	5
4	Namespace Documentation	7
4.1	Package Core	7
4.2	Package hello	8
4.3	Package Parser	9
4.4	Package security	10
5	Class Documentation	11
5.1	Parser.ConectorWeb Class Reference	11
5.1.1	Detailed Description	11
5.1.2	Member Function Documentation	11
5.1.2.1	sendHttpGet	11
5.2	Core.File Class Reference	12
5.2.1	Detailed Description	12
5.2.2	Constructor & Destructor Documentation	12
5.2.2.1	File	12

5.2.2.2	File	13
5.2.3	Member Function Documentation	13
5.2.3.1	getCreationdate	13
5.2.3.2	getFilename	13
5.2.3.3	getFileSize	14
5.2.3.4	getId	14
5.2.3.5	getLicenseCollection	14
5.2.3.6	getLocalpath	14
5.2.3.7	getModificationdate	14
5.2.3.8	setCreationdate	15
5.2.3.9	setFilename	15
5.2.3.10	setFileSize	15
5.2.3.11	setId	15
5.2.3.12	setLicenseCollection	16
5.2.3.13	setLocalpath	16
5.2.3.14	setModificationdate	16
5.2.3.15	toString	16
5.2.3.16	writeFile	16
5.3	Core.FileContent Class Reference	18
5.3.1	Detailed Description	18
5.3.2	Constructor & Destructor Documentation	18
5.3.2.1	FileContent	18
5.3.2.2	FileContent	18
5.3.3	Member Function Documentation	19
5.3.3.1	getContent	19
5.3.3.2	getFileid	19
5.3.3.3	getId	19
5.3.3.4	setContent	19
5.3.3.5	setFileid	20
5.3.3.6	setId	20
5.4	hello.HelloMIDlet Class Reference	21

5.4.1	Detailed Description	24
5.4.2	Constructor & Destructor Documentation	24
5.4.2.1	HelloMIDlet	24
5.4.3	Member Function Documentation	24
5.4.3.1	commandAction	24
5.4.3.2	destroyApp	24
5.4.3.3	exitMIDlet	25
5.4.3.4	getAccesoCorrecto	25
5.4.3.5	getAccesoIncorrecto	25
5.4.3.6	getAlertaArchivos	25
5.4.3.7	getAlertaArchivosCS	26
5.4.3.8	getBackCommand	26
5.4.3.9	getBackCommand1	26
5.4.3.10	getBackCommand10	26
5.4.3.11	getBackCommand2	26
5.4.3.12	getBackCommand3	27
5.4.3.13	getBackCommand4	27
5.4.3.14	getBackCommand5	27
5.4.3.15	getBackCommand6	27
5.4.3.16	getBackCommand7	28
5.4.3.17	getBackCommand8	28
5.4.3.18	getBackCommand9	28
5.4.3.19	getBitacoraCreadaDown	28
5.4.3.20	getBitacoraCreadaUp	28
5.4.3.21	getBitacoraErrorDown	29
5.4.3.22	getBitacoraErrorUp	29
5.4.3.23	getCancelCommand	29
5.4.3.24	getCancelCommand1	29
5.4.3.25	getCancelCommand2	30
5.4.3.26	getCancelCommand25	30
5.4.3.27	getCancelCommand30	30

5.4.3.28	getChoiceGroup	30
5.4.3.29	getChoiceGroup1	30
5.4.3.30	getChoiceGroup2	31
5.4.3.31	getChoiceGroup3	31
5.4.3.32	getChoiceGroup4	31
5.4.3.33	getChoiceGroup5	31
5.4.3.34	getDisplay	32
5.4.3.35	getExitCommand	32
5.4.3.36	getExitCommand1	32
5.4.3.37	getExitCommand2	32
5.4.3.38	getFileBrowser	32
5.4.3.39	getForm	33
5.4.3.40	getForm1	33
5.4.3.41	getForm2	33
5.4.3.42	getForm3	33
5.4.3.43	getForm4	34
5.4.3.44	getForm5	34
5.4.3.45	getForm6	34
5.4.3.46	getImage1	34
5.4.3.47	getImage2	34
5.4.3.48	getImage3	35
5.4.3.49	getItemCommand	35
5.4.3.50	getItemCommand1	35
5.4.3.51	getItemCommand2	35
5.4.3.52	getItemCommand3	36
5.4.3.53	getItemCommand4	36
5.4.3.54	getItemCommand5	36
5.4.3.55	getList	36
5.4.3.56	getList2	36
5.4.3.57	getList3	37
5.4.3.58	getLogirScreen	37

5.4.3.59	getOkCommand	37
5.4.3.60	getOkCommand1	37
5.4.3.61	getOkCommand10	38
5.4.3.62	getOkCommand11	38
5.4.3.63	getOkCommand12	38
5.4.3.64	getOkCommand13	38
5.4.3.65	getOkCommand14	38
5.4.3.66	getOkCommand15	39
5.4.3.67	getOkCommand16	39
5.4.3.68	getOkCommand17	39
5.4.3.69	getOkCommand18	39
5.4.3.70	getOkCommand19	40
5.4.3.71	getOkCommand2	40
5.4.3.72	getOkCommand20	40
5.4.3.73	getOkCommand21	40
5.4.3.74	getOkCommand22	40
5.4.3.75	getOkCommand23	41
5.4.3.76	getOkCommand24	41
5.4.3.77	getOkCommand25	41
5.4.3.78	getOkCommand3	41
5.4.3.79	getOkCommand30	42
5.4.3.80	getOkCommand4	42
5.4.3.81	getOkCommand5	42
5.4.3.82	getOkCommand6	42
5.4.3.83	getOkCommand7	42
5.4.3.84	getOkCommand8	43
5.4.3.85	getOkCommand9	43
5.4.3.86	getProcesoAcceso	43
5.4.3.87	getSincCorrecto	43
5.4.3.88	getSincError	44
5.4.3.89	getSplashScreen	44

5.4.3.90	getTask	44
5.4.3.91	getTask1	44
5.4.3.92	getTask2	44
5.4.3.93	getTask3	45
5.4.3.94	getTask4	45
5.4.3.95	getTask5	45
5.4.3.96	getTask6	45
5.4.3.97	getTask7	46
5.4.3.98	getTextBox	46
5.4.3.99	getTextBox1	46
5.4.3.100	getTextBox5	46
5.4.3.101	getTextBox6	46
5.4.3.102	getTextField	47
5.4.3.103	getTicker	47
5.4.3.104	getTickerSalida	47
5.4.3.105	getTransferenciaCompletaDown	47
5.4.3.106	getTransferenciaCompletaUp	47
5.4.3.107	getTransferenciaErrorDown	48
5.4.3.108	getTransferenciaErrorUp	48
5.4.3.109	getVerArchivosMovil	48
5.4.3.110	getVerArchivosServidor	48
5.4.3.111	getWaitScreen	48
5.4.3.112	getWaitScreen1	49
5.4.3.113	getWaitScreen2	49
5.4.3.114	getWaitScreen3	49
5.4.3.115	getWaitScreen4	49
5.4.3.116	getWaitScreen5	50
5.4.3.117	getWaitScreen6	50
5.4.3.118	list2Action	50
5.4.3.119	list3Action	50
5.4.3.120	listAction	50

5.4.3.121	method	50
5.4.3.122	method1	51
5.4.3.123	pauseApp	51
5.4.3.124	readLine	51
5.4.3.125	resumeMIDlet	51
5.4.3.126	startApp	51
5.4.3.127	startMIDlet	51
5.4.3.128	switchDisplayable	51
5.5	Core.License Class Reference	53
5.5.1	Detailed Description	53
5.5.2	Constructor & Destructor Documentation	53
5.5.2.1	License	53
5.5.2.2	License	53
5.5.3	Member Function Documentation	54
5.5.3.1	getFileid	54
5.5.3.2	getId	54
5.5.3.3	getType	54
5.5.3.4	getUserid	54
5.5.3.5	setFileid	55
5.5.3.6	setId	55
5.5.3.7	setType	55
5.5.3.8	setuserid	55
5.5.3.9	toString	55
5.6	Core.Propiedades Class Reference	57
5.6.1	Detailed Description	57
5.6.2	Constructor & Destructor Documentation	57
5.6.2.1	Propiedades	57
5.6.2.2	Propiedades	57
5.6.3	Member Function Documentation	57
5.6.3.1	getUrlserveridor	57
5.6.3.2	setUrlserveridor	58

5.7	Parser.RestParser Class Reference	59
5.7.1	Detailed Description	59
5.7.2	Member Function Documentation	59
5.7.2.1	beginParseFile	59
5.7.2.2	beginParseFile	60
5.7.2.3	beginParseFileContent	60
5.7.2.4	beginParseFileContent	60
5.7.2.5	beginParseFiles	60
5.7.2.6	beginParseLicense	61
5.7.2.7	beginParseLicense	61
5.7.2.8	beginParseLicenses	61
5.7.2.9	beginParsePropiedad	62
5.7.2.10	beginParsePropiedad	62
5.7.2.11	beginParseUser	62
5.7.2.12	beginParseUser	62
5.7.2.13	beginParseUsers	63
5.7.2.14	getDocument	63
5.7.2.15	getDocumentLocal	63
5.7.2.16	setDocumentLocal	63
5.8	security.Sha1 Class Reference	65
5.8.1	Detailed Description	65
5.8.2	Member Function Documentation	65
5.8.2.1	getHash	65
5.9	Parser.StringTokenizer Class Reference	66
5.9.1	Detailed Description	67
5.9.2	Constructor & Destructor Documentation	67
5.9.2.1	StringTokenizer	67
5.9.2.2	StringTokenizer	68
5.9.2.3	StringTokenizer	68
5.9.2.4	StringTokenizer	69
5.9.2.5	StringTokenizer	69

5.9.3	Member Function Documentation	70
5.9.3.1	countTokens	70
5.9.3.2	countTokens	71
5.9.3.3	countTokens	71
5.9.3.4	countTokens	72
5.9.3.5	countTokens	72
5.9.3.6	getCurrentPosition	73
5.9.3.7	hasMoreElements	73
5.9.3.8	hasMoreTokens	73
5.9.3.9	hasNext	74
5.9.3.10	next	74
5.9.3.11	nextElement	75
5.9.3.12	nextToken	75
5.9.3.13	nextToken	76
5.9.3.14	nextToken	77
5.9.3.15	nextToken	77
5.9.3.16	nextToken	78
5.9.3.17	peek	78
5.9.3.18	remove	79
5.9.3.19	restOfText	79
5.9.3.20	setDelimiters	80
5.9.3.21	setDelimiters	80
5.9.3.22	setDelimiters	80
5.9.3.23	setDelimiters	81
5.9.3.24	setReturnEmptyTokens	81
5.9.3.25	setText	82
5.9.3.26	skipDelimiters	83
5.9.3.27	toArray	83
5.9.4	Member Data Documentation	83
5.9.4.1	delimsChangedPosition	83
5.9.4.2	emptyReturned	84

5.9.4.3	maxDelimChar	84
5.9.4.4	nontokenDelims	84
5.9.4.5	position	85
5.9.4.6	returnEmptyTokens	85
5.9.4.7	strLength	85
5.9.4.8	text	85
5.9.4.9	tokenCount	86
5.9.4.10	tokenDelims	86
5.10	Parser.UnsupportedOperationException Class Reference	87
5.10.1	Detailed Description	87
5.11	Core.User Class Reference	88
5.11.1	Detailed Description	88
5.11.2	Constructor & Destructor Documentation	88
5.11.2.1	User	88
5.11.2.2	User	88
5.11.3	Member Function Documentation	89
5.11.3.1	getId	89
5.11.3.2	getLicenseColletion	89
5.11.3.3	getPassword	89
5.11.3.4	getUsername	89
5.11.3.5	setId	90
5.11.3.6	setLicenseColletion	90
5.11.3.7	setPassword	90
5.11.3.8	setUsername	90
5.11.3.9	toString	90
6	File Documentation	93
6.1	src/Core/File.java File Reference	93
6.2	src/Core/File.java	94
6.3	src/Core/FileContent.java File Reference	97
6.4	src/Core/FileContent.java	98

6.5	src/Core/License.java File Reference	99
6.6	src/Core/License.java	100
6.7	src/Core/Propiedades.java File Reference	102
6.8	src/Core/Propiedades.java	103
6.9	src/Core/User.java File Reference	104
6.10	src/Core/User.java	105
6.11	src/hello/HelloMIDlet.java File Reference	107
6.12	src/hello/HelloMIDlet.java	108
6.13	src/Parser/ConectorWeb.java File Reference	192
6.14	src/Parser/ConectorWeb.java	193
6.15	src/Parser/RestParser.java File Reference	194
6.16	src/Parser/RestParser.java	195
6.17	src/Parser/StringTokenizer.java File Reference	204
6.18	src/Parser/StringTokenizer.java	205
6.19	src/security/Sha1.java File Reference	215
6.20	src/security/Sha1.java	216

Chapter 1

Namespace Index

1.1 Package List

Here are the packages with brief descriptions (if available):

Core	7
hello	8
Parser	9
security	10

Chapter 2

Class Index

2.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Parser.ConectorWeb	11
Core.File	12
Core.FileContent	18
hello.HelloMIDlet	21
Core.License	53
Core.Propiedades	57
Parser.RestParser	59
security.Sha1	65
Parser.StringTokenizer	66
Parser.UnsupportedOperationException	87
Core.User	88

Chapter 3

File Index

3.1 File List

Here is a list of all files with brief descriptions:

src/Core/File.java	93
src/Core/FileContent.java	97
src/Core/License.java	99
src/Core/Propiedades.java	102
src/Core/User.java	104
src/hello/HelloMIDlet.java	107
src/Parser/ConectorWeb.java	192
src/Parser/RestParser.java	194
src/Parser/StringTokenizer.java	204
src/security/Sha1.java	215

Chapter 4

Namespace Documentation

4.1 Package Core

Classes

- class [File](#)
- class [FileContent](#)
- class [License](#)
- class [Propiedades](#)
- class [User](#)

4.2 Package hello

Classes

- class [HelloMIDlet](#)

4.3 Package Parser

Classes

- class [ConectorWeb](#)
- class [RestParser](#)
- class [StringTokenizer](#)
- class [UnsupportedOperationException](#)

4.4 Package security

Classes

- class [Sha1](#)

Chapter 5

Class Documentation

5.1 Parser.ConectorWeb Class Reference

Static Public Member Functions

- static String [sendHttpGet](#) (String url) throws IOException

5.1.1 Detailed Description

Author

Rodrigo Ramírez Reyes

Definition at line 18 of file [ConectorWeb.java](#).

5.1.2 Member Function Documentation

5.1.2.1 static String Parser.ConectorWeb.sendHttpGet (String url) throws IOException [static]

Definition at line 21 of file [ConectorWeb.java](#).

The documentation for this class was generated from the following file:

- [src/Parser/ConectorWeb.java](#)

5.2 Core.File Class Reference

Public Member Functions

- [File](#) ()
- [File](#) (Integer id, String filename, int filesize, String localpath, Date creationdate, Date modificationdate, Vector licenseCollection)
- Date [getCreationdate](#) ()
- void [setCreationdate](#) (Date creationdate)
- String [getFilename](#) ()
- void [setFilename](#) (String filename)
- int [getFilesize](#) ()
- void [setFilesize](#) (int filesize)
- Integer [getId](#) ()
- void [setId](#) (Integer id)
- Vector [getLicenseCollection](#) ()
- void [setLicenseCollection](#) (Vector licenseCollection)
- String [getLocalpath](#) ()
- void [setLocalpath](#) (String localpath)
- Date [getModificationdate](#) ()
- void [setModificationdate](#) (Date modificationdate)
- String [toString](#) ()

Static Public Member Functions

- static boolean [writeFile](#) (String path, byte[] data)

5.2.1 Detailed Description

Author

Rodrigo Ramírez Reyes

Definition at line 19 of file [File.java](#).

5.2.2 Constructor & Destructor Documentation

5.2.2.1 Core.File.File ()

Función constructor vacío.

Definition at line 41 of file [File.java](#).

5.2.2.2 Core.File.File (Integer *id*, String *filename*, int *filesize*, String *localpath*, Date *creationdate*, Date *modificationdate*, Vector *licenseCollection*)

Constructor del objeto [File](#).

Parameters

- id* Identificador del archivo
- filename* Nombre del archivo
- filesize* Tamaño del archivo
- localpath* Path del archivo
- creationdate* Fecha de creación
- modificationdate* Fecha de modificación
- licenseCollection* Arreglo de permisos sobre el archivo

Definition at line 56 of file [File.java](#).

5.2.3 Member Function Documentation

5.2.3.1 Date Core.File.getCreationdate ()

Función para obtener la fecha de creación de un archivo

Returns

Regresa la fecha de creación de un archivo

Definition at line 73 of file [File.java](#).

5.2.3.2 String Core.File.getFilename ()

Función para obtener el nombre del archivo

Returns

Regresa el nombre de un archivo

Definition at line 93 of file [File.java](#).

5.2.3.3 `int Core.File.getFilesize ()`

Función para obtener el tamaño de un archivo

Returns

Regresa el tamaño de un archivo

Definition at line 113 of file [File.java](#).

5.2.3.4 `Integer Core.File.getId ()`

Función para obtener el Id de un archivo

Returns

Regresa el Id de un archivo

Definition at line 133 of file [File.java](#).

5.2.3.5 `Vector Core.File.getLicenseCollection ()`

Función para obtener el arreglo de permisos de un archivo

Returns

Regresa el arreglo de permisos de un archivo

Definition at line 153 of file [File.java](#).

5.2.3.6 `String Core.File.getLocalpath ()`

Función para obtener la ruta de un archivo

Returns

Regresa la ruta de un archivo

Definition at line 173 of file [File.java](#).

5.2.3.7 `Date Core.File.getModificationdate ()`

Función para obtener la fecha de modificación de un archivo

Returns

Regresa la fecha de modificación de un archivo

Definition at line 193 of file [File.java](#).

5.2.3.8 void Core.File.setCreationdate (Date *creationdate*)

Función para fijar la fecha de creación de un archivo

Parameters

creationdate Una fecha de tpo Date

Definition at line 83 of file [File.java](#).

5.2.3.9 void Core.File.setFilename (String *filename*)

Función para fijar el nombre de un archivo

Parameters

filename Una nombre de tipo String

Definition at line 103 of file [File.java](#).

5.2.3.10 void Core.File.setFilesize (int *filesize*)

Función para fijar el tamaño de un archivo

Parameters

filesize Tamaño de un archivo de tipo entero

Definition at line 123 of file [File.java](#).

5.2.3.11 void Core.File.setId (Integer *id*)

Función para fijar el Id de un archivo

Parameters

id Id de un archivo de tipo Integer

Definition at line 143 of file [File.java](#).

5.2.3.12 void Core.File.setLicenseColletion (Vector *licenseColletion*)

Función para fijar el arreglo de permisos de un archivo

Parameters

licenseColletion Arreglo de permisos de un archivo

Definition at line 163 of file [File.java](#).

5.2.3.13 void Core.File.setLocalpath (String *localpath*)

Función para fijar la ruta de un archivo

Parameters

localpath Ruta de un archivo de tpo String

Definition at line 183 of file [File.java](#).

5.2.3.14 void Core.File.setModificationdate (Date *modificationdate*)

Función para fijar la fecha de modificación de un archivo

Parameters

modificationdate Fecha de modificación de un archivo de tipo Date

Definition at line 203 of file [File.java](#).

5.2.3.15 String Core.File.toString ()

Función que sobrescribe al método toString

Returns

Una cadena con todos los parámetros

Definition at line 213 of file [File.java](#).

**5.2.3.16 static boolean Core.File.writeFile (String *path*, byte[] *data*)
[static]**

Función para la creación de archivos de forma local (en el móvil)

Parameters

path Una ruta del archivo de tipo String

data Un arreglo de datos de un archivo

Returns

Regresa un valor booleano. True si se ha creado con éxito, false en caso contrario

Definition at line 233 of file [File.java](#).

The documentation for this class was generated from the following file:

- [src/Core/File.java](#)

5.3 Core.FileContent Class Reference

Public Member Functions

- [FileContent \(\)](#)
- [FileContent \(Integer id, String fileid, byte\[\] content\)](#)
- [byte\[\] getContent \(\)](#)
- [void setContent \(byte\[\] content\)](#)
- [String getFileid \(\)](#)
- [void setFileid \(String fileid\)](#)
- [Integer getId \(\)](#)
- [void setId \(Integer id\)](#)

5.3.1 Detailed Description

Author

Rodrigo Ramírez Reyes

Definition at line 12 of file [FileContent.java](#).

5.3.2 Constructor & Destructor Documentation

5.3.2.1 Core.FileContent.FileContent ()

Función constructor vacío.

Definition at line 29 of file [FileContent.java](#).

5.3.2.2 Core.FileContent.FileContent (Integer id, String fileid, byte[] content)

Constructor del objeto [FileContent](#).

Parameters

id Id de un filecontent de tipo Integer

fileid Id del contenido de un archivo de tipo String

content Arreglo con el contenido de un archivo de tipo byte[]

Definition at line 40 of file [FileContent.java](#).

5.3.3 Member Function Documentation

5.3.3.1 byte [] Core.FileContent.getContent ()

Función para obtener el contenido de un archivo

Returns

Arreglo con el contenido de un archivo

Definition at line 52 of file [FileContent.java](#).

5.3.3.2 String Core.FileContent.getFileid ()

Función para obtener el id de un archivo

Returns

Regresa un id de un archivo

Definition at line 72 of file [FileContent.java](#).

5.3.3.3 Integer Core.FileContent.getId ()

Función para obtener el id de un [FileContent](#)

Returns

Id de un [FileContent](#)

Definition at line 92 of file [FileContent.java](#).

5.3.3.4 void Core.FileContent.setContent (byte[] content)

Función para fijar el contenido de un archivo

Parameters

content Arreglo de un archivo de tipo byte[]

Definition at line 62 of file [FileContent.java](#).

5.3.3.5 void Core.FileContent.setFileid (String *fileid*)

Función para fijar el id de un [FileContent](#)

Parameters

fileid Id de un [FileContent](#) de tipo String

Definition at line 82 of file [FileContent.java](#).

5.3.3.6 void Core.FileContent.setId (Integer *id*)

Función para fijar el Id de un [FileContent](#)

Parameters

id Id de un [FileContent](#) de tipo Integer

Definition at line 102 of file [FileContent.java](#).

The documentation for this class was generated from the following file:

- [src/Core/FileContent.java](#)

5.4 hello.HelloMIDlet Class Reference

Public Member Functions

- [HelloMIDlet \(\)](#)
- void [startMIDlet \(\)](#)
- void [resumeMIDlet \(\)](#)
- void [switchDisplayable](#) (Alert alert, Displayable nextDisplayable)
- void [commandAction](#) (Command command, Displayable displayable)
- Command [getExitCommand \(\)](#)
- LoginScreen [getLoginScreen \(\)](#)
- WaitScreen [getProcesoAcceso \(\)](#)
- Alert [getAccesoIncorrecto \(\)](#)
- Alert [getAccesoCorrecto \(\)](#)
- SimpleCancellableTask [getTask \(\)](#)
- Command [getOkCommand \(\)](#)
- Command [getOkCommand1 \(\)](#)
- List [getList \(\)](#)
- void [listAction \(\)](#)
- Command [getOkCommand2 \(\)](#)
- Command [getOkCommand3 \(\)](#)
- Command [getOkCommand4 \(\)](#)
- Command [getOkCommand5 \(\)](#)
- FileBrowser [getFileBrowser \(\)](#)
- Command [getBackCommand \(\)](#)
- Command [getOkCommand6 \(\)](#)
- Command [getBackCommand1 \(\)](#)
- Command [getExitCommand1 \(\)](#)
- List [getList2 \(\)](#)
- void [list2Action \(\)](#)
- List [getList3 \(\)](#)
- void [list3Action \(\)](#)
- Command [getBackCommand2 \(\)](#)
- Command [getBackCommand3 \(\)](#)
- Command [getBackCommand4 \(\)](#)
- WaitScreen [getWaitScreen \(\)](#)
- SimpleCancellableTask [getTask1 \(\)](#)
- Command [getOkCommand7 \(\)](#)
- Alert [getBitacoraCreadaUp \(\)](#)
- Alert [getBitacoraErrorUp \(\)](#)
- Command [getOkCommand8 \(\)](#)
- Command [getBackCommand5 \(\)](#)

- Command [getBackCommand6 \(\)](#)
- Command [getOkCommand9 \(\)](#)
- Command [getOkCommand10 \(\)](#)
- Command [getOkCommand11 \(\)](#)
- Command [getOkCommand12 \(\)](#)
- TextBox [getTextBox1 \(\)](#)
- Command [getOkCommand13 \(\)](#)
- WaitScreen [getWaitScreen1 \(\)](#)
- Alert [getTransferenciaCompletaUp \(\)](#)
- SimpleCancellableTask [getTask2 \(\)](#)
- Alert [getTransferenciaErrorUp \(\)](#)
- Command [getOkCommand14 \(\)](#)
- WaitScreen [getWaitScreen2 \(\)](#)
- Alert [getBitacoraCreadaDown \(\)](#)
- Alert [getBitacoraErrorDown \(\)](#)
- TextBox [getTextBox5 \(\)](#)
- TextBox [getAlertaArchivos \(\)](#)
- Command [getOkCommand25 \(\)](#)
- Command [getCancelCommand25 \(\)](#)
- TextBox [getAlertaArchivosCS \(\)](#)
- Command [getOkCommand30 \(\)](#)
- Command [getCancelCommand30 \(\)](#)
- SimpleCancellableTask [getTask3 \(\)](#)
- Command [getOkCommand15 \(\)](#)
- WaitScreen [getWaitScreen3 \(\)](#)
- Alert [getTransferenciaCompletaDown \(\)](#)
- Alert [getTransferenciaErrorDown \(\)](#)
- SimpleCancellableTask [getTask4 \(\)](#)
- Form [getForm \(\)](#)
- ChoiceGroup [getChoiceGroup \(\)](#)
- Command [getOkCommand16 \(\)](#)
- Command [getOkCommand17 \(\)](#)
- Form [getForm1 \(\)](#)
- ChoiceGroup [getChoiceGroup1 \(\)](#)
- Command [getBackCommand7 \(\)](#)
- Command [getBackCommand8 \(\)](#)
- Command [getItemCommand \(\)](#)
- Command [getItemCommand1 \(\)](#)
- Command [getBackCommand9 \(\)](#)
- Form [getForm2 \(\)](#)
- TextField [getTextField \(\)](#)
- TextBox [getTextBox6 \(\)](#)
- Command [getOkCommand18 \(\)](#)

- Form [getForm3](#) ()
- Command [getOkCommand19](#) ()
- ChoiceGroup [getChoiceGroup2](#) ()
- Command [getOkCommand20](#) ()
- WaitScreen [getWaitScreen4](#) ()
- SimpleCancellableTask [getTask5](#) ()
- Command [getCancelCommand](#) ()
- SplashScreen [getSplashScreen](#) ()
- Command [getExitCommand2](#) ()
- Image [getImage1](#) ()
- Command [getCancelCommand1](#) ()
- Command [getOkCommand21](#) ()
- Form [getForm4](#) ()
- WaitScreen [getWaitScreen5](#) ()
- SimpleCancellableTask [getTask6](#) ()
- ChoiceGroup [getChoiceGroup3](#) ()
- Alert [getVerArchivosMovil](#) ()
- Alert [getVerArchivosServidor](#) ()
- Image [getImage2](#) ()
- Image [getImage3](#) ()
- Command [getItemCommand2](#) ()
- Command [getItemCommand3](#) ()
- Command [getItemCommand4](#) ()
- Command [getItemCommand5](#) ()
- Command [getOkCommand22](#) ()
- Command [getBackCommand10](#) ()
- Form [getForm5](#) ()
- ChoiceGroup [getChoiceGroup4](#) ()
- WaitScreen [getWaitScreen6](#) ()
- SimpleCancellableTask [getTask7](#) ()
- Command [getOkCommand23](#) ()
- Form [getForm6](#) ()
- ChoiceGroup [getChoiceGroup5](#) ()
- void [method](#) ()
- void [method1](#) ()
- Command [getOkCommand24](#) ()
- TextBox [getTextBox](#) ()
- Command [getCancelCommand2](#) ()
- Alert [getSincCorrecto](#) ()
- Alert [getSincError](#) ()
- Display [getDisplay](#) ()
- void [exitMIDlet](#) ()
- void [startApp](#) ()

- void `pauseApp ()`
- void `destroyApp (boolean unconditional)`
- Ticker `getTicker ()`
- Ticker `getTickerSalida ()`

Package Functions

- String `readLine (Reader r)` throws Exception

5.4.1 Detailed Description

Author

Rodrigo Ramírez Reyes

Definition at line 44 of file `HelloMIDlet.java`.

5.4.2 Constructor & Destructor Documentation

5.4.2.1 `hello.HelloMIDlet.HelloMIDlet ()`

The `HelloMIDlet` constructor.

Definition at line 199 of file `HelloMIDlet.java`.

5.4.3 Member Function Documentation

5.4.3.1 `void hello.HelloMIDlet.commandAction (Command command, Displayable displayable)`

Called by a system to indicated that a command has been invoked on a particular displayable.

Parameters

command the Command that was invoked

displayable the Displayable where the command was invoked

Definition at line 263 of file `HelloMIDlet.java`.

5.4.3.2 `void hello.HelloMIDlet.destroyApp (boolean unconditional)`

Called to signal the MIDlet to terminate.

Parameters

unconditional if true, then the MIDlet has to be unconditionally terminated and all resources has to be released.

Definition at line 3111 of file [HelloMIDlet.java](#).

5.4.3.3 void hello.HelloMIDlet.exitMIDlet ()

Exits MIDlet.

Definition at line 3080 of file [HelloMIDlet.java](#).

5.4.3.4 Alert hello.HelloMIDlet.getAccesoCorrecto ()

Returns an initalized instance of accesoCorrecto component.

Returns

the initialized component instance

Definition at line 874 of file [HelloMIDlet.java](#).

5.4.3.5 Alert hello.HelloMIDlet.getAccesoIncorrecto ()

Returns an initalized instance of accesoIncorrecto component.

Returns

the initialized component instance

Definition at line 857 of file [HelloMIDlet.java](#).

5.4.3.6 TextBox hello.HelloMIDlet.getAlertaArchivos ()

Regresa una alerta para la confirmacion de archivos.

Returns

regresa el objeto de la alerta

Definition at line 1711 of file [HelloMIDlet.java](#).

5.4.3.7 TextBox `hello.HelloMIDlet.getAlertaArchivosCS ()`

Regresa una alerta para la confirmacion de archivos.

Returns

regresa el objeto de la alerta

Definition at line 1766 of file [HelloMIDlet.java](#).

5.4.3.8 Command `hello.HelloMIDlet.getBackCommand ()`

Returns an initalized instance of backCommand component.

Returns

the initialized component instance

Definition at line 1091 of file [HelloMIDlet.java](#).

5.4.3.9 Command `hello.HelloMIDlet.getBackCommand1 ()`

Returns an initalized instance of backCommand1 component.

Returns

the initialized component instance

Definition at line 1125 of file [HelloMIDlet.java](#).

5.4.3.10 Command `hello.HelloMIDlet.getBackCommand10 ()`

Returns an initalized instance of backCommand10 component.

Returns

the initialized component instance

Definition at line 2673 of file [HelloMIDlet.java](#).

5.4.3.11 Command `hello.HelloMIDlet.getBackCommand2 ()`

Returns an initalized instance of backCommand2 component.

Returns

the initialized component instance

Definition at line 1265 of file [HelloMIDlet.java](#).

5.4.3.12 Command hello.HelloMIDlet.getBackCommand3 ()

Returns an initialized instance of backCommand3 component.

Returns

the initialized component instance

Definition at line 1280 of file [HelloMIDlet.java](#).

5.4.3.13 Command hello.HelloMIDlet.getBackCommand4 ()

Returns an initialized instance of backCommand4 component.

Returns

the initialized component instance

Definition at line 1299 of file [HelloMIDlet.java](#).

5.4.3.14 Command hello.HelloMIDlet.getBackCommand5 ()

Returns an initialized instance of backCommand5 component.

Returns

the initialized component instance

Definition at line 1424 of file [HelloMIDlet.java](#).

5.4.3.15 Command hello.HelloMIDlet.getBackCommand6 ()

Returns an initialized instance of backCommand6 component.

Returns

the initialized component instance

Definition at line 1439 of file [HelloMIDlet.java](#).

5.4.3.16 Command `hello.HelloMIDlet.getBackCommand7 ()`

Returns an initialized instance of backCommand7 component.

Returns

the initialized component instance

Definition at line 2052 of file [HelloMIDlet.java](#).

5.4.3.17 Command `hello.HelloMIDlet.getBackCommand8 ()`

Returns an initialized instance of backCommand8 component.

Returns

the initialized component instance

Definition at line 2067 of file [HelloMIDlet.java](#).

5.4.3.18 Command `hello.HelloMIDlet.getBackCommand9 ()`

Returns an initialized instance of backCommand9 component.

Returns

the initialized component instance

Definition at line 2118 of file [HelloMIDlet.java](#).

5.4.3.19 Alert `hello.HelloMIDlet.getBitacoraCreadaDown ()`

Returns an initialized instance of bitacoraCreadaDown component.

Returns

the initialized component instance

Definition at line 1660 of file [HelloMIDlet.java](#).

5.4.3.20 Alert `hello.HelloMIDlet.getBitacoraCreadaUp ()`

Returns an initialized instance of bitacoraCreadaUp component.

Returns

the initialized component instance

Definition at line 1371 of file [HelloMIDlet.java](#).

5.4.3.21 Alert hello.HelloMIDlet.getBitacoraErrorDown ()

Returns an initialized instance of bitacoraErrorDown component.

Returns

the initialized component instance

Definition at line 1677 of file [HelloMIDlet.java](#).

5.4.3.22 Alert hello.HelloMIDlet.getBitacoraErrorUp ()

Returns an initialized instance of bitacoraErrorUp component.

Returns

the initialized component instance

Definition at line 1388 of file [HelloMIDlet.java](#).

5.4.3.23 Command hello.HelloMIDlet.getCancelCommand ()

Returns an initialized instance of cancelCommand component.

Returns

the initialized component instance

Definition at line 2334 of file [HelloMIDlet.java](#).

5.4.3.24 Command hello.HelloMIDlet.getCancelCommand1 ()

Returns an initialized instance of cancelCommand1 component.

Returns

the initialized component instance

Definition at line 2403 of file [HelloMIDlet.java](#).

5.4.3.25 Command `hello.HelloMIDlet.getCancelCommand2 ()`

Returns an initialized instance of `cancelCommand2` component.

Returns

the initialized component instance

Definition at line 3003 of file [HelloMIDlet.java](#).

5.4.3.26 Command `hello.HelloMIDlet.getCancelCommand25 ()`

Returns an initialized instance of `cancelCommand` component.

Returns

the initialized component instance

Definition at line 1752 of file [HelloMIDlet.java](#).

5.4.3.27 Command `hello.HelloMIDlet.getCancelCommand30 ()`

Returns an initialized instance of `cancelCommand` component.

Returns

the initialized component instance

Definition at line 1810 of file [HelloMIDlet.java](#).

5.4.3.28 ChoiceGroup `hello.HelloMIDlet.getChoiceGroup ()`

Returns an initialized instance of `choiceGroup` component.

Returns

the initialized component instance

Definition at line 1960 of file [HelloMIDlet.java](#).

5.4.3.29 ChoiceGroup `hello.HelloMIDlet.getChoiceGroup1 ()`

Returns an initialized instance of `choiceGroup1` component.

Returns

the initialized component instance

Definition at line 2033 of file [HelloMIDlet.java](#).

5.4.3.30 ChoiceGroup hello.HelloMIDlet.getChoiceGroup2 ()

Returns an initialized instance of choiceGroup2 component.

Returns

the initialized component instance

Definition at line 2241 of file [HelloMIDlet.java](#).

5.4.3.31 ChoiceGroup hello.HelloMIDlet.getChoiceGroup3 ()

Returns an initialized instance of choiceGroup3 component.

Returns

the initialized component instance

Definition at line 2491 of file [HelloMIDlet.java](#).

5.4.3.32 ChoiceGroup hello.HelloMIDlet.getChoiceGroup4 ()

Returns an initialized instance of choiceGroup4 component.

Returns

the initialized component instance

Definition at line 2706 of file [HelloMIDlet.java](#).

5.4.3.33 ChoiceGroup hello.HelloMIDlet.getChoiceGroup5 ()

Returns an initialized instance of choiceGroup5 component.

Returns

the initialized component instance

Definition at line 2906 of file [HelloMIDlet.java](#).

5.4.3.34 Display `hello.HelloMIDlet.getDisplay ()`

Returns a display instance.

Returns

the display instance.

Definition at line 3072 of file [HelloMIDlet.java](#).

5.4.3.35 Command `hello.HelloMIDlet.getExitCommand ()`

Returns an initialized instance of exitCommand component.

Returns

the initialized component instance

Definition at line 794 of file [HelloMIDlet.java](#).

5.4.3.36 Command `hello.HelloMIDlet.getExitCommand1 ()`

Returns an initialized instance of exitCommand1 component.

Returns

the initialized component instance

Definition at line 1140 of file [HelloMIDlet.java](#).

5.4.3.37 Command `hello.HelloMIDlet.getExitCommand2 ()`

Returns an initialized instance of exitCommand2 component.

Returns

the initialized component instance

Definition at line 2369 of file [HelloMIDlet.java](#).

5.4.3.38 FileBrowser `hello.HelloMIDlet.getFileBrowser ()`

Returns an initialized instance of fileBrowser component.

Returns

the initialized component instance

Definition at line 1072 of file [HelloMIDlet.java](#).

5.4.3.39 Form hello.HelloMIDlet.getForm ()

Returns an initialized instance of form component.

Returns

the initialized component instance

Definition at line 1941 of file [HelloMIDlet.java](#).

5.4.3.40 Form hello.HelloMIDlet.getForm1 ()

Returns an initialized instance of form1 component.

Returns

the initialized component instance

Definition at line 2015 of file [HelloMIDlet.java](#).

5.4.3.41 Form hello.HelloMIDlet.getForm2 ()

Returns an initialized instance of form2 component.

Returns

the initialized component instance

Definition at line 2133 of file [HelloMIDlet.java](#).

5.4.3.42 Form hello.HelloMIDlet.getForm3 ()

Returns an initialized instance of form3 component.

Returns

the initialized component instance

Definition at line 2207 of file [HelloMIDlet.java](#).

5.4.3.43 Form `hello.HelloMIDlet.getForm4 ()`

Returns an initialized instance of form4 component.

Returns

the initialized component instance

Definition at line 2433 of file [HelloMIDlet.java](#).

5.4.3.44 Form `hello.HelloMIDlet.getForm5 ()`

Returns an initialized instance of form5 component.

Returns

the initialized component instance

Definition at line 2688 of file [HelloMIDlet.java](#).

5.4.3.45 Form `hello.HelloMIDlet.getForm6 ()`

Returns an initialized instance of form6 component.

Returns

the initialized component instance

Definition at line 2889 of file [HelloMIDlet.java](#).

5.4.3.46 Image `hello.HelloMIDlet.getImage1 ()`

Returns an initialized instance of image1 component.

Returns

the initialized component instance

Definition at line 2384 of file [HelloMIDlet.java](#).

5.4.3.47 Image `hello.HelloMIDlet.getImage2 ()`

Returns an initialized instance of image2 component.

Returns

the initialized component instance

Definition at line 2560 of file [HelloMIDlet.java](#).

5.4.3.48 Image hello.HelloMIDlet.getImage3 ()

Returns an initialized instance of image3 component.

Returns

the initialized component instance

Definition at line 2579 of file [HelloMIDlet.java](#).

5.4.3.49 Command hello.HelloMIDlet.getItemCommand ()

Returns an initialized instance of itemCommand component.

Returns

the initialized component instance

Definition at line 2086 of file [HelloMIDlet.java](#).

5.4.3.50 Command hello.HelloMIDlet.getItemCommand1 ()

Returns an initialized instance of itemCommand1 component.

Returns

the initialized component instance

Definition at line 2103 of file [HelloMIDlet.java](#).

5.4.3.51 Command hello.HelloMIDlet.getItemCommand2 ()

Returns an initialized instance of itemCommand2 component.

Returns

the initialized component instance

Definition at line 2598 of file [HelloMIDlet.java](#).

5.4.3.52 Command `hello.HelloMIDlet.getItemCommand3 ()`

Returns an initialized instance of `itemCommand3` component.

Returns

the initialized component instance

Definition at line 2613 of file [HelloMIDlet.java](#).

5.4.3.53 Command `hello.HelloMIDlet.getItemCommand4 ()`

Returns an initialized instance of `itemCommand4` component.

Returns

the initialized component instance

Definition at line 2628 of file [HelloMIDlet.java](#).

5.4.3.54 Command `hello.HelloMIDlet.getItemCommand5 ()`

Returns an initialized instance of `itemCommand5` component.

Returns

the initialized component instance

Definition at line 2643 of file [HelloMIDlet.java](#).

5.4.3.55 List `hello.HelloMIDlet.getList ()`

Returns an initialized instance of `list` component.

Returns

the initialized component instance

Definition at line 944 of file [HelloMIDlet.java](#).

5.4.3.56 List `hello.HelloMIDlet.getList2 ()`

Returns an initialized instance of `list2` component.

Returns

the initialized component instance

Definition at line 1155 of file [HelloMIDlet.java](#).

5.4.3.57 List hello.HelloMIDlet.getList3 ()

Returns an initialized instance of list3 component.

Returns

the initialized component instance

Definition at line 1197 of file [HelloMIDlet.java](#).

5.4.3.58 LoginScreen hello.HelloMIDlet.getLoginScreen ()

Returns an initialized instance of loginScreen component.

Returns

the initialized component instance

Definition at line 810 of file [HelloMIDlet.java](#).

5.4.3.59 Command hello.HelloMIDlet.getOkCommand ()

Returns an initialized instance of okCommand component.

Returns

the initialized component instance

Definition at line 914 of file [HelloMIDlet.java](#).

5.4.3.60 Command hello.HelloMIDlet.getOkCommand1 ()

Returns an initialized instance of okCommand1 component.

Returns

the initialized component instance

Definition at line 929 of file [HelloMIDlet.java](#).

5.4.3.61 Command `hello.HelloMIDlet.getOkCommand10 ()`

Returns an initialized instance of `okCommand10` component.

Returns

the initialized component instance

Definition at line 1469 of file [HelloMIDlet.java](#).

5.4.3.62 Command `hello.HelloMIDlet.getOkCommand11 ()`

Returns an initialized instance of `okCommand11` component.

Returns

the initialized component instance

Definition at line 1484 of file [HelloMIDlet.java](#).

5.4.3.63 Command `hello.HelloMIDlet.getOkCommand12 ()`

Returns an initialized instance of `okCommand12` component.

Returns

the initialized component instance

Definition at line 1501 of file [HelloMIDlet.java](#).

5.4.3.64 Command `hello.HelloMIDlet.getOkCommand13 ()`

Returns an initialized instance of `okCommand13` component.

Returns

the initialized component instance

Definition at line 1534 of file [HelloMIDlet.java](#).

5.4.3.65 Command `hello.HelloMIDlet.getOkCommand14 ()`

Returns an initialized instance of `okCommand14` component.

Returns

the initialized component instance

Definition at line 1625 of file [HelloMIDlet.java](#).

5.4.3.66 Command hello.HelloMIDlet.getOkCommand15 ()

Returns an initialized instance of okCommand15 component.

Returns

the initialized component instance

Definition at line 1846 of file [HelloMIDlet.java](#).

5.4.3.67 Command hello.HelloMIDlet.getOkCommand16 ()

Returns an initialized instance of okCommand16 component.

Returns

the initialized component instance

Definition at line 1983 of file [HelloMIDlet.java](#).

5.4.3.68 Command hello.HelloMIDlet.getOkCommand17 ()

Returns an initialized instance of okCommand17 component.

Returns

the initialized component instance

Definition at line 2000 of file [HelloMIDlet.java](#).

5.4.3.69 Command hello.HelloMIDlet.getOkCommand18 ()

Returns an initialized instance of okCommand18 component.

Returns

the initialized component instance

Definition at line 2192 of file [HelloMIDlet.java](#).

5.4.3.70 Command `hello.HelloMIDlet.getOkCommand19 ()`

Returns an initialized instance of `okCommand19` component.

Returns

the initialized component instance

Definition at line 2226 of file [HelloMIDlet.java](#).

5.4.3.71 Command `hello.HelloMIDlet.getOkCommand2 ()`

Returns an initialized instance of `okCommand2` component.

Returns

the initialized component instance

Definition at line 1004 of file [HelloMIDlet.java](#).

5.4.3.72 Command `hello.HelloMIDlet.getOkCommand20 ()`

Returns an initialized instance of `okCommand20` component.

Returns

the initialized component instance

Definition at line 2277 of file [HelloMIDlet.java](#).

5.4.3.73 Command `hello.HelloMIDlet.getOkCommand21 ()`

Returns an initialized instance of `okCommand21` component.

Returns

the initialized component instance

Definition at line 2418 of file [HelloMIDlet.java](#).

5.4.3.74 Command `hello.HelloMIDlet.getOkCommand22 ()`

Returns an initialized instance of `okCommand22` component.

Returns

the initialized component instance

Definition at line 2658 of file [HelloMIDlet.java](#).

5.4.3.75 Command hello.HelloMIDlet.getOkCommand23 ()

Returns an initialized instance of okCommand23 component.

Returns

the initialized component instance

Definition at line 2874 of file [HelloMIDlet.java](#).

5.4.3.76 Command hello.HelloMIDlet.getOkCommand24 ()

Returns an initialized instance of okCommand24 component.

Returns

the initialized component instance

Definition at line 2967 of file [HelloMIDlet.java](#).

5.4.3.77 Command hello.HelloMIDlet.getOkCommand25 ()

Returns an initialized instance of okCommand15 component.

Returns

the initialized component instance

Definition at line 1739 of file [HelloMIDlet.java](#).

5.4.3.78 Command hello.HelloMIDlet.getOkCommand3 ()

Returns an initialized instance of okCommand3 component.

Returns

the initialized component instance

Definition at line 1021 of file [HelloMIDlet.java](#).

5.4.3.79 Command `hello.HelloMIDlet.getOkCommand30 ()`

Returns an initialized instance of `okCommand15` component.

Returns

the initialized component instance

Definition at line 1796 of file [HelloMIDlet.java](#).

5.4.3.80 Command `hello.HelloMIDlet.getOkCommand4 ()`

Returns an initialized instance of `okCommand4` component.

Returns

the initialized component instance

Definition at line 1036 of file [HelloMIDlet.java](#).

5.4.3.81 Command `hello.HelloMIDlet.getOkCommand5 ()`

Returns an initialized instance of `okCommand5` component.

Returns

the initialized component instance

Definition at line 1057 of file [HelloMIDlet.java](#).

5.4.3.82 Command `hello.HelloMIDlet.getOkCommand6 ()`

Returns an initialized instance of `okCommand6` component.

Returns

the initialized component instance

Definition at line 1106 of file [HelloMIDlet.java](#).

5.4.3.83 Command `hello.HelloMIDlet.getOkCommand7 ()`

Returns an initialized instance of `okCommand7` component.

Returns

the initialized component instance

Definition at line 1356 of file [HelloMIDlet.java](#).

5.4.3.84 Command hello.HelloMIDlet.getOkCommand8 ()

Returns an initialized instance of okCommand8 component.

Returns

the initialized component instance

Definition at line 1409 of file [HelloMIDlet.java](#).

5.4.3.85 Command hello.HelloMIDlet.getOkCommand9 ()

Returns an initialized instance of okCommand9 component.

Returns

the initialized component instance

Definition at line 1454 of file [HelloMIDlet.java](#).

5.4.3.86 WaitScreen hello.HelloMIDlet.getProcesoAcceso ()

Returns an initialized instance of procesoAcceso component.

Returns

the initialized component instance

Definition at line 839 of file [HelloMIDlet.java](#).

5.4.3.87 Alert hello.HelloMIDlet.getSincCorrecto ()

Returns an initialized instance of sincCorrecto component.

Returns

the initialized component instance

Definition at line 3018 of file [HelloMIDlet.java](#).

5.4.3.88 Alert `hello.HelloMIDlet.getSincError ()`

Returns an initialized instance of SincError component.

Returns

the initialized component instance

Definition at line 3034 of file [HelloMIDlet.java](#).

5.4.3.89 SplashScreen `hello.HelloMIDlet.getSplashScreen ()`

Returns an initialized instance of splashScreen component.

Returns

the initialized component instance

Definition at line 2349 of file [HelloMIDlet.java](#).

5.4.3.90 SimpleCancelableTask `hello.HelloMIDlet.getTask ()`

Returns an initialized instance of task component.

Returns

the initialized component instance

Definition at line 891 of file [HelloMIDlet.java](#).

5.4.3.91 SimpleCancelableTask `hello.HelloMIDlet.getTask1 ()`

Returns an initialized instance of task1 component.

Returns

the initialized component instance

Definition at line 1334 of file [HelloMIDlet.java](#).

5.4.3.92 SimpleCancelableTask `hello.HelloMIDlet.getTask2 ()`

Returns an initialized instance of task2 component.

Returns

the initialized component instance

Definition at line 1584 of file [HelloMIDlet.java](#).

5.4.3.93 SimpleCancelableTask hello.HelloMIDlet.getTask3 ()

Returns an initialized instance of task3 component.

Returns

the initialized component instance

Definition at line 1824 of file [HelloMIDlet.java](#).

5.4.3.94 SimpleCancelableTask hello.HelloMIDlet.getTask4 ()

Returns an initialized instance of task4 component.

Returns

the initialized component instance

Definition at line 1916 of file [HelloMIDlet.java](#).

5.4.3.95 SimpleCancelableTask hello.HelloMIDlet.getTask5 ()

Returns an initialized instance of task5 component.

Returns

the initialized component instance

Definition at line 2313 of file [HelloMIDlet.java](#).

5.4.3.96 SimpleCancelableTask hello.HelloMIDlet.getTask6 ()

Returns an initialized instance of task6 component.

Returns

the initialized component instance

Definition at line 2470 of file [HelloMIDlet.java](#).

5.4.3.97 SimpleCancelableTask `hello.HelloMIDlet.getTask7 ()`

Returns an initialized instance of task7 component.

Returns

the initialized component instance

Definition at line 2748 of file [HelloMIDlet.java](#).

5.4.3.98 TextBox `hello.HelloMIDlet.getTextBox ()`

Returns an initialized instance of textBox component.

Returns

the initialized component instance

Definition at line 2982 of file [HelloMIDlet.java](#).

5.4.3.99 TextBox `hello.HelloMIDlet.getTextBox1 ()`

Returns an initialized instance of textBox1 component.

Returns

the initialized component instance

Definition at line 1516 of file [HelloMIDlet.java](#).

5.4.3.100 TextBox `hello.HelloMIDlet.getTextBox5 ()`

Returns an initialized instance of textBox5 component.

Returns

the initialized component instance

Definition at line 1694 of file [HelloMIDlet.java](#).

5.4.3.101 TextBox `hello.HelloMIDlet.getTextBox6 ()`

Returns an initialized instance of textBox6 component.

Returns

the initialized component instance

Definition at line 2175 of file [HelloMIDlet.java](#).

5.4.3.102 TextField hello.HelloMIDlet.getTextField ()

Returns an initialized instance of textField component.

Returns

the initialized component instance

Definition at line 2154 of file [HelloMIDlet.java](#).

5.4.3.103 Ticker hello.HelloMIDlet.getTicker ()

Definition at line 3115 of file [HelloMIDlet.java](#).

5.4.3.104 Ticker hello.HelloMIDlet.getTickerSalida ()

Definition at line 3131 of file [HelloMIDlet.java](#).

5.4.3.105 Alert hello.HelloMIDlet.getTransferenciaCompletaDown ()

Returns an initialized instance of transferenciaCompletaDown component.

Returns

the initialized component instance

Definition at line 1879 of file [HelloMIDlet.java](#).

5.4.3.106 Alert hello.HelloMIDlet.getTransferenciaCompletaUp ()

Returns an initialized instance of transferenciaCompletaUp component.

Returns

the initialized component instance

Definition at line 1567 of file [HelloMIDlet.java](#).

5.4.3.107 Alert hello.HelloMIDlet.getTransferenciaErrorDown ()

Returns an initialized instance of transferenciaErrorDown component.

Returns

the initialized component instance

Definition at line 1899 of file [HelloMIDlet.java](#).

5.4.3.108 Alert hello.HelloMIDlet.getTransferenciaErrorUp ()

Returns an initialized instance of transferenciaErrorUp component.

Returns

the initialized component instance

Definition at line 1608 of file [HelloMIDlet.java](#).

5.4.3.109 Alert hello.HelloMIDlet.getVerArchivosMovil ()

Returns an initialized instance of verArchivosMovil component.

Returns

the initialized component instance

Definition at line 2528 of file [HelloMIDlet.java](#).

5.4.3.110 Alert hello.HelloMIDlet.getVerArchivosServidor ()

Returns an initialized instance of verArchivosServidor component.

Returns

the initialized component instance

Definition at line 2544 of file [HelloMIDlet.java](#).

5.4.3.111 WaitScreen hello.HelloMIDlet.getWaitScreen ()

Returns an initialized instance of waitScreen component.

Returns

the initialized component instance

Definition at line 1314 of file [HelloMIDlet.java](#).

5.4.3.112 WaitScreen hello.HelloMIDlet.getWaitScreen1 ()

Returns an initialized instance of waitScreen1 component.

Returns

the initialized component instance

Definition at line 1549 of file [HelloMIDlet.java](#).

5.4.3.113 WaitScreen hello.HelloMIDlet.getWaitScreen2 ()

Returns an initialized instance of waitScreen2 component.

Returns

the initialized component instance

Definition at line 1642 of file [HelloMIDlet.java](#).

5.4.3.114 WaitScreen hello.HelloMIDlet.getWaitScreen3 ()

Returns an initialized instance of waitScreen3 component.

Returns

the initialized component instance

Definition at line 1861 of file [HelloMIDlet.java](#).

5.4.3.115 WaitScreen hello.HelloMIDlet.getWaitScreen4 ()

Returns an initialized instance of waitScreen4 component.

Returns

the initialized component instance

Definition at line 2294 of file [HelloMIDlet.java](#).

5.4.3.116 WaitScreen hello.HelloMIDlet.getWaitScreen5 ()

Returns an initialized instance of waitScreen5 component.

Returns

the initialized component instance

Definition at line 2452 of file [HelloMIDlet.java](#).

5.4.3.117 WaitScreen hello.HelloMIDlet.getWaitScreen6 ()

Returns an initialized instance of waitScreen6 component.

Returns

the initialized component instance

Definition at line 2730 of file [HelloMIDlet.java](#).

5.4.3.118 void hello.HelloMIDlet.list2Action ()

Performs an action assigned to the selected list element in the list2 component.

Definition at line 1174 of file [HelloMIDlet.java](#).

5.4.3.119 void hello.HelloMIDlet.list3Action ()

Performs an action assigned to the selected list element in the list3 component.

Definition at line 1253 of file [HelloMIDlet.java](#).

5.4.3.120 void hello.HelloMIDlet.listAction ()

Performs an action assigned to the selected list element in the list component.

Definition at line 968 of file [HelloMIDlet.java](#).

5.4.3.121 void hello.HelloMIDlet.method ()

Performs an action assigned to the method if-point.

Definition at line 2924 of file [HelloMIDlet.java](#).

5.4.3.122 void hello.HelloMIDlet.method1 ()

Performs an action assigned to the method1 if-point.

Definition at line 2944 of file [HelloMIDlet.java](#).

5.4.3.123 void hello.HelloMIDlet.pauseApp ()

Called when MIDlet is paused.

Definition at line 3103 of file [HelloMIDlet.java](#).

**5.4.3.124 String hello.HelloMIDlet.readLine (Reader *r*) throws Exception
[package]**

Definition at line 4086 of file [HelloMIDlet.java](#).

5.4.3.125 void hello.HelloMIDlet.resumeMIDlet ()

Performs an action assigned to the Mobile Device - MIDlet Resumed point.

Definition at line 232 of file [HelloMIDlet.java](#).

5.4.3.126 void hello.HelloMIDlet.startApp ()

Called when MIDlet is started. Checks whether the MIDlet have been already started and initialize/starts or resumes the MIDlet.

Definition at line 3090 of file [HelloMIDlet.java](#).

5.4.3.127 void hello.HelloMIDlet.startMIDlet ()

Performs an action assigned to the Mobile Device - MIDlet Started point.

Definition at line 221 of file [HelloMIDlet.java](#).

**5.4.3.128 void hello.HelloMIDlet.switchDisplayable (Alert *alert*, Displayable
nextDisplayable)**

Switches a current displayable in a display. The `display` instance is taken from `getDisplay` method. This method is used by all actions in the design for switching displayable.

Parameters

alert the Alert which is temporarily set to the display; if `null`, then `nextDisplayable` is set immediately

nextDisplayable the Displayable to be set

Definition at line 245 of file [HelloMIDlet.java](#).

The documentation for this class was generated from the following file:

- [src/hello/HelloMIDlet.java](#)

5.5 Core.License Class Reference

Public Member Functions

- [License \(\)](#)
- [License \(Integer id, String type, String userid, String fileid\)](#)
- [String getFileid \(\)](#)
- [void setFileid \(String fileid\)](#)
- [Integer getId \(\)](#)
- [void setId \(Integer id\)](#)
- [String getType \(\)](#)
- [void setType \(String type\)](#)
- [String getUserid \(\)](#)
- [void setUserid \(String userid\)](#)
- [String toString \(\)](#)

5.5.1 Detailed Description

Author

Rodrigo Ramírez Reyes

Definition at line 12 of file [License.java](#).

5.5.2 Constructor & Destructor Documentation

5.5.2.1 Core.License.License ()

Función constructor vacío.

Definition at line 30 of file [License.java](#).

5.5.2.2 Core.License.License (Integer *id*, String *type*, String *userid*, String *fileid*)

Constructor del objeto [License](#).

Parameters

- id* Id de la licencia de tipo Integer
- type* Tipo de licencia de tipo String
- userid* Id del usuario con esta licencia de tipo String
- fileid* Id del archivo con esta licencia de tipo String

Definition at line 42 of file [License.java](#).

5.5.3 Member Function Documentation

5.5.3.1 String Core.License.getFileid ()

Función para obtener el Id de una licencia

Returns

Regresa el Id de una licencia

Definition at line 55 of file [License.java](#).

5.5.3.2 Integer Core.License.getId ()

Función para obtener el Id de una licencia

Returns

Regresa el Id de una licencia

Definition at line 75 of file [License.java](#).

5.5.3.3 String Core.License.getType ()

Función para obtener el tipo de licencia de tipo String

Returns

Regresa el tipo de licencia

Definition at line 95 of file [License.java](#).

5.5.3.4 String Core.License.getUserid ()

Función para obtener el Id de una licencia

Returns

Regresa el Id de una licencia

Definition at line 115 of file [License.java](#).

5.5.3.5 void Core.License.setFileid (String *fileid*)

Función que fija el Id de una licencia

Parameters

fileid Id de una licencia de tipo String

Definition at line 65 of file [License.java](#).

5.5.3.6 void Core.License.setId (Integer *id*)

Función para fijar el Id de una licencia

Parameters

id Id de una licencia de tipo Integer

Definition at line 85 of file [License.java](#).

5.5.3.7 void Core.License.setType (String *type*)

Función para fijar el tipo de una licencia

Parameters

type Tipo de una licencia de tipo String

Definition at line 105 of file [License.java](#).

5.5.3.8 void Core.License.setUserid (String *userid*)

Función para fijar el Id de una licencia

Parameters

userid Id de una licencia de tipo String

Definition at line 125 of file [License.java](#).

5.5.3.9 String Core.License.toString ()

Función que sobrescribe al método toString

Returns

Una cadena con todos los parámetros

Definition at line 135 of file [License.java](#).

The documentation for this class was generated from the following file:

- [src/Core/License.java](#)

5.6 Core.Propiedades Class Reference

Public Member Functions

- [Propiedades \(\)](#)
- [Propiedades \(String urlservidor\)](#)
- [String getUrlservidor \(\)](#)
- [void setUrlservidor \(String urlservidor\)](#)

5.6.1 Detailed Description

Author

Rodrigo Ramírez Reyes

Definition at line 12 of file [Propiedades.java](#).

5.6.2 Constructor & Destructor Documentation

5.6.2.1 Core.Propiedades.Propiedades ()

Función constructor vacío.

Definition at line 29 of file [Propiedades.java](#).

5.6.2.2 Core.Propiedades.Propiedades (String urlservidor)

Parameters

urlservidor Es la url predefinida por el usuario para conectarse al servidor

Definition at line 40 of file [Propiedades.java](#).

5.6.3 Member Function Documentation

5.6.3.1 String Core.Propiedades.getUrlservidor ()

Función para obtener la url del servidor

Returns

url de tipo String

Definition at line 50 of file [Propiedades.java](#).

5.6.3.2 void `Core.Propiedades.setUrlservidor` (String *urlservidor*)

Función para fijar el url de un servidor

Parameters

urlservidor

Definition at line 61 of file `Propiedades.java`.

The documentation for this class was generated from the following file:

- `src/Core/Propiedades.java`

5.7 Parser.RestParser Class Reference

Static Public Member Functions

- static [User](#) [beginParseUser](#) (Element root)
- static [User](#) [beginParseUser](#) (Document doc)
- static [Vector](#) [beginParseUsers](#) (Document doc)
- static [License](#) [beginParseLicense](#) (Element root)
- static [License](#) [beginParseLicense](#) (Document doc)
- static [Vector](#) [beginParseLicenses](#) (Document doc)
- static [File](#) [beginParseFile](#) (Element root)
- static [File](#) [beginParseFile](#) (Document doc)
- static [Vector](#) [beginParseFiles](#) (Document doc)
- static [Document](#) [getDocument](#) (String url)
- static [Document](#) [getDocumentLocal](#) (String url)
- static void [setDocumentLocal](#) (String url)
- static [FileContent](#) [beginParseFileContent](#) (Element root)
- static [FileContent](#) [beginParseFileContent](#) (Document doc)
- static [Propiedades](#) [beginParsePropiedad](#) (Element root)
- static [Propiedades](#) [beginParsePropiedad](#) (Document doc)

5.7.1 Detailed Description

Author

Rodrigo Ramírez Reyes

Definition at line 34 of file [RestParser.java](#).

5.7.2 Member Function Documentation

5.7.2.1 static File [Parser.RestParser.beginParseFile](#) (Document *doc*) [static]

Función que parsea un archivo desde un Document doc

Parameters

doc Elemento doc desde donde se comenzará a parsear

Returns

Regresa un objeto File

Definition at line 279 of file [RestParser.java](#).

5.7.2.2 static File Parser.RestParser.beginParseFile (Element *root*) [static]

Función que parsea un archivo desde un Element root

Parameters

root Elemento raíz desde donde se comenzará a parsear

Returns

Regresa un objeto File

Definition at line 223 of file [RestParser.java](#).

5.7.2.3 static FileContent Parser.RestParser.beginParseFileContent (Document *doc*) [static]

Función que parsea un FileContent desde un Document doc

Parameters

doc Elemento doc desde donde se comenzará a parsear

Returns

Regresa un objeto FileContent

Definition at line 504 of file [RestParser.java](#).

5.7.2.4 static FileContent Parser.RestParser.beginParseFileContent (Element *root*) [static]

Función que parsea un FileContent desde un Element root

Parameters

root Elemento raíz desde donde se comenzará a parsear

Returns

Regresa un objeto FileContent

Definition at line 461 of file [RestParser.java](#).

5.7.2.5 static Vector Parser.RestParser.beginParseFiles (Document *doc*) [static]

Definition at line 286 of file [RestParser.java](#).

5.7.2.6 static License Parser.RestParser.beginParseLicense (Document *doc*) [static]

Función para parsear una Licencia desde un Document *doc*

Parameters

doc Elemento *doc* desde donde se comenzará a parsear

Returns

Regresa un objeto Licencia

Definition at line 175 of file RestParser.java.

5.7.2.7 static License Parser.RestParser.beginParseLicense (Element *root*) [static]

Función para parsear una Licencia por medio de un Elemento *root*

Parameters

root Elemento raíz desde donde se comenzará a parsear

Returns

regresa un objeto Licencia

Definition at line 131 of file RestParser.java.

5.7.2.8 static Vector Parser.RestParser.beginParseLicenses (Document *doc*) [static]

Función que parsea una serie de Licencias desde un Element *doc*

Parameters

doc Elemento *doc* desde donde se comenzará a parsear

Returns

Regresa un arreglo de objetos Licencia

Definition at line 187 of file RestParser.java.

5.7.2.9 static Propiedades Parser.RestParser.beginParsePropiedad (Document *doc*) [static]

Función para parsear un usuario por medio de un Document *doc*

Parameters

doc Elemento documento desde donde se comenzará a parsear

Returns

Regresa un objeto Usuario

Definition at line 541 of file [RestParser.java](#).

5.7.2.10 static Propiedades Parser.RestParser.beginParsePropiedad (Element *root*) [static]

Definition at line 511 of file [RestParser.java](#).

5.7.2.11 static User Parser.RestParser.beginParseUser (Document *doc*) [static]

Función para parsear un usuario por medio de un Document *doc*

Parameters

doc Elemento documento desde donde se comenzará a parsear

Returns

Regresa un objeto Usuario

Definition at line 89 of file [RestParser.java](#).

5.7.2.12 static User Parser.RestParser.beginParseUser (Element *root*) [static]

Función para parsear un usuario por medio de un Element *root*

Parameters

root Elemento raíz desde donde se comenzará a parsear

Returns

Regresa un objeto Usuario

Definition at line 43 of file [RestParser.java](#).

5.7.2.13 static Vector Parser.RestParser.beginParseUsers (Document doc) [static]

Definition at line 96 of file [RestParser.java](#).

5.7.2.14 static Document Parser.RestParser.getDocument (String url) [static]

Función que parsea un XML desde una ruta de destino

Parameters

url Ruta del XML a obtener de tipo String

Returns

Regresa un Document doc con el contenido de un XML

Definition at line 313 of file [RestParser.java](#).

5.7.2.15 static Document Parser.RestParser.getDocumentLocal (String url) [static]

Función que parsea un XML desde un archivo local

Parameters

url Ruta del XML a obtener de tipo String

Returns

Regresa un Document doc con el contenido de un XML

Definition at line 362 of file [RestParser.java](#).

5.7.2.16 static void Parser.RestParser.setDocumentLocal (String url) [static]

Función que escribe un contenido XML desde un archivo local

Parameters

url Ruta del XML a obtener de tipo String

Returns

Regresa un Document doc con el contenido de un XML

Definition at line 412 of file [RestParser.java](#).

The documentation for this class was generated from the following file:

- [src/Parser/RestParser.java](#)

5.8 security.Sha1 Class Reference

Public Member Functions

- String [getHash](#) (String message) throws NoSuchAlgorithmException

5.8.1 Detailed Description

Author

Rodrigo Ramírez Reyes

Definition at line 22 of file [Sha1.java](#).

5.8.2 Member Function Documentation

5.8.2.1 String security.Sha1.getHash (String message) throws NoSuchAlgorithmException

Función para obtener el Hash (SHA-1) de una cadena de texto

Parameters

message String con una cadena de texto

Returns

Regresa una cadena en formato SHA-1

Exceptions

NoSuchAlgorithmException En caso de algún error, regresa la excepción de que no existe el algoritmo solicitado.

Definition at line 50 of file [Sha1.java](#).

The documentation for this class was generated from the following file:

- [src/security/Sha1.java](#)

5.9 Parser.StringTokenizer Class Reference

Public Member Functions

- [StringTokenizer](#) (String text, String nontokenDelims, String tokenDelims)
- [StringTokenizer](#) (String text, String nontokenDelims, String tokenDelims, boolean returnEmptyTokens)
- [StringTokenizer](#) (String text, String delims, boolean delimsAreTokens)
- [StringTokenizer](#) (String text, String nontokenDelims)
- [StringTokenizer](#) (String text)
- void [setText](#) (String text)
- boolean [hasMoreTokens](#) ()
- String [nextToken](#) ()
- boolean [skipDelimiters](#) ()
- int [countTokens](#) ()
- void [setDelimiters](#) (String delims)
- void [setDelimiters](#) (String delims, boolean delimsAreTokens)
- void [setDelimiters](#) (String nontokenDelims, String tokenDelims)
- void [setDelimiters](#) (String nontokenDelims, String tokenDelims, boolean returnEmptyTokens)
- int [countTokens](#) (String delims)
- int [countTokens](#) (String delims, boolean delimsAreTokens)
- int [countTokens](#) (String nontokenDelims, String tokenDelims)
- int [countTokens](#) (String nontokenDelims, String tokenDelims, boolean returnEmptyTokens)
- String [nextToken](#) (String nontokenDelims, String tokenDelims)
- String [nextToken](#) (String nontokenDelims, String tokenDelims, boolean returnEmptyTokens)
- String [nextToken](#) (String delims, boolean delimsAreTokens)
- String [nextToken](#) (String nontokenDelims)
- boolean [hasMoreElements](#) ()
- Object [nextElement](#) ()
- boolean [hasNext](#) ()
- Object [next](#) ()
- void [remove](#) ()
- void [setReturnEmptyTokens](#) (boolean returnEmptyTokens)
- int [getCurrentPosition](#) ()
- String[] [toArray](#) ()
- String [restOfText](#) ()
- String [peek](#) ()

Protected Attributes

- String *text*
- int *strLength*
- String *nontokenDelims*
- String *tokenDelims*
- int *position*
- boolean *emptyReturned*
- char *maxDelimChar*
- boolean *returnEmptyTokens*
- int *delimsChangedPosition*
- int *tokenCount*

5.9.1 Detailed Description

Definition at line 27 of file [StringTokenizer.java](#).

5.9.2 Constructor & Destructor Documentation

5.9.2.1 Parser.StringTokenizer.StringTokenizer (String *text*, String *nontokenDelims*, String *tokenDelims*)

Constructs a string tokenizer for the specified string. Both token and nontoken delimiters are specified.

The current position is set at the beginning of the string.

Parameters

text a string to be parsed.

nontokenDelims the nontoken delimiters, i.e. the delimiters that only separate tokens and are not returned as separate tokens.

tokenDelims the token delimiters, i.e. delimiters that both separate tokens, and are themselves returned as tokens.

Exceptions

NullPointerException if *text* is null.

Since

ostermillerutils 1.00.00

Definition at line 149 of file [StringTokenizer.java](#).

5.9.2.2 `Parser.StringTokenizer.StringTokenizer` (`String text`, `String nontokenDelims`, `String tokenDelims`, `boolean returnEmptyTokens`)

Constructs a string tokenizer for the specified string. Both token and nontoken delimiters are specified and whether or not empty tokens are returned is specified.

Empty tokens are tokens that are between consecutive delimiters.

It is a primary constructor (i.e. all other constructors are defined in terms of it.)

The current position is set at the beginning of the string.

Parameters

text a string to be parsed.

nontokenDelims the nontoken delimiters, i.e. the delimiters that only separate tokens and are not returned as separate tokens.

tokenDelims the token delimiters, i.e. delimiters that both separate tokens, and are themselves returned as tokens.

returnEmptyTokens true if empty tokens may be returned; false otherwise.

Exceptions

NullPointerException if text is null.

Since

ostermillerutils 1.00.00

Definition at line 175 of file [StringTokenizer.java](#).

5.9.2.3 `Parser.StringTokenizer.StringTokenizer` (`String text`, `String delims`, `boolean delimsAreTokens`)

Constructs a string tokenizer for the specified string. Either token or nontoken delimiters are specified.

Is equivalent to:

- If the third parameter is false -- `StringTokenizer(text, delims, null)`
- If the third parameter is true -- `StringTokenizer(text, null, delims)`

Parameters

text a string to be parsed.

delims the delimiters.

delimsAreTokens flag indicating whether the second parameter specifies token or nontoken delimiters: `false` -- the second parameter specifies nontoken delimiters, the set of token delimiters is empty; `true` -- the second parameter specifies token delimiters, the set of nontoken delimiters is empty.

Exceptions

NullPointerException if text is null.

Since

ostermillerutils 1.00.00

Definition at line 205 of file [StringTokenizer.java](#).

5.9.2.4 Parser.StringTokenizer.StringTokenizer (String text, String nontokenDelims)

Constructs a string tokenizer for the specified string. The characters in the `nontokenDelims` argument are the delimiters for separating tokens. Delimiter characters themselves will not be treated as tokens.

Is equivalent to `StringTokenizer(text, nontokenDelims, null)`.

Parameters

text a string to be parsed.

nontokenDelims the nontoken delimiters.

Exceptions

NullPointerException if text is null.

Since

ostermillerutils 1.00.00

Definition at line 222 of file [StringTokenizer.java](#).

5.9.2.5 Parser.StringTokenizer.StringTokenizer (String text)

Constructs a string tokenizer for the specified string. The tokenizer uses "`\t\n\r\f`" as a delimiter set of nontoken delimiters, and an empty token delimiter set.

Is equivalent to `StringTokenizer(text, "\t\n\r\f", null)`;

Parameters

text a string to be parsed.

Exceptions

NullPointerException if *text* is null.

Since

ostermillerutils 1.00.00

Definition at line 238 of file [StringTokenizer.java](#).

5.9.3 Member Function Documentation

5.9.3.1 `int Parser.StringTokenizer.countTokens (String nontokenDelims, String tokenDelims, boolean returnEmptyTokens)`

Calculates the number of times that this tokenizer's `nextToken` method can be called before it generates an exception using the given set of delimiters. The delimiters given will be used for future calls to `nextToken()` unless new delimiters are given. The current position is not advanced.

Parameters

nontokenDelims the new set of nontoken delimiters.

tokenDelims the new set of token delimiters.

returnEmptyTokens true if empty tokens may be returned; false otherwise.

Returns

the number of tokens remaining in the string using the new delimiter set.

See also

[countTokens\(\)](#)

Since

ostermillerutils 1.00.00

Definition at line 609 of file [StringTokenizer.java](#).

5.9.3.2 `int Parser.StringTokenizer.countTokens (String nontokenDelims, String tokenDelims)`

Calculates the number of times that this tokenizer's `nextToken` method can be called before it generates an exception using the given set of delimiters. The delimiters given will be used for future calls to `nextToken()` unless new delimiters are given. The current position is not advanced.

Parameters

nontokenDelims the new set of nontoken delimiters.

tokenDelims the new set of token delimiters.

Returns

the number of tokens remaining in the string using the new delimiter set.

See also

[countTokens\(\)](#)

Since

ostermillerutils 1.00.00

Definition at line 588 of file [StringTokenizer.java](#).

5.9.3.3 `int Parser.StringTokenizer.countTokens (String delims, boolean delimsAreTokens)`

Calculates the number of times that this tokenizer's `nextToken` method can be called before it generates an exception using the given set of delimiters. The delimiters given will be used for future calls to `nextToken()` unless new delimiters are given. The current position is not advanced.

Parameters

delims the new set of delimiters.

delimsAreTokens flag indicating whether the first parameter specifies token or nontoken delimiters: false -- the first parameter specifies nontoken delimiters, the set of token delimiters is empty; true -- the first parameter specifies token delimiters, the set of nontoken delimiters is empty.

Returns

the number of tokens remaining in the string using the new delimiter set.

See also[countTokens\(\)](#)**Since**

ostermillerutils 1.00.00

Definition at line 568 of file [StringTokenizer.java](#).**5.9.3.4 int Parser.StringTokenizer.countTokens (String delims)**

Calculates the number of times that this tokenizer's `nextToken` method can be called before it generates an exception using the given set of (nontoken) delimiters. The delimiters given will be used for future calls to `nextToken()` unless new delimiters are given. The current position is not advanced.

Parameters

delims the new set of nontoken delimiters (the set of token delimiters will be empty).

Returns

the number of tokens remaining in the string using the new delimiter set.

See also[countTokens\(\)](#)**Since**

ostermillerutils 1.00.00

Definition at line 545 of file [StringTokenizer.java](#).**5.9.3.5 int Parser.StringTokenizer.countTokens ()**

Calculates the number of times that this tokenizer's `nextToken` method can be called before it generates an exception. The current position is not advanced.

Returns

the number of tokens remaining in the string using the current delimiter set.

See also[nextToken\(\)](#)

Since

ostermillerutils 1.00.00

Definition at line 438 of file [StringTokenizer.java](#).

5.9.3.6 int Parser.StringTokenizer.getCurrentPosition ()

Get the the index of the character immediately following the end of the last token. This is the position at which this tokenizer will begin looking for the next token when a [nextToken\(\)](#) method is invoked.

Returns

the current position or -1 if the entire string has been tokenized.

Since

ostermillerutils 1.00.00

Definition at line 952 of file [StringTokenizer.java](#).

5.9.3.7 boolean Parser.StringTokenizer.hasMoreElements ()

Returns the same value as the [hasMoreTokens\(\)](#) method. It exists so that this class can implement the `Enumeration` interface.

Returns

`true` if there are more tokens; `false` otherwise.

See also

[java.util.Enumeration](#)
[hasMoreTokens\(\)](#)

Since

ostermillerutils 1.00.00

Definition at line 850 of file [StringTokenizer.java](#).

5.9.3.8 boolean Parser.StringTokenizer.hasMoreTokens ()

Tests if there are more tokens available from this tokenizer's string. If this method returns `true`, then a subsequent call to [nextToken](#) with no argument will successfully return a token.

The current position is not changed.

Returns

`true` if and only if there is at least one token in the string after the current position;
`false` otherwise.

Since

ostermillerutils 1.00.00

Definition at line 320 of file [StringTokenizer.java](#).

5.9.3.9 boolean Parser.StringTokenizer.hasNext ()

Returns the same value as the `hasMoreTokens ()` method. It exists so that this class can implement the `Iterator` interface.

Returns

`true` if there are more tokens; `false` otherwise.

See also

[java.util.Iterator](#)
[hasMoreTokens\(\)](#)

Since

ostermillerutils 1.00.00

Definition at line 882 of file [StringTokenizer.java](#).

5.9.3.10 Object Parser.StringTokenizer.next ()

Returns the same value as the `nextToken ()` method, except that its declared return value is `Object` rather than `String`. It exists so that this class can implement the `Iterator` interface.

Returns

the next token in the string.

Exceptions

NoSuchElementException if there are no more tokens in this tokenizer's string.

See also

[java.util.Iterator](#)
[nextToken\(\)](#)

Since

ostermillerutils 1.00.00

Definition at line 899 of file [StringTokenizer.java](#).

5.9.3.11 Object Parser.StringTokenizer.nextElement ()

Returns the same value as the `nextToken()` method, except that its declared return value is `Object` rather than `String`. It exists so that this class can implement the `Enumeration` interface.

Returns

the next token in the string.

Exceptions

NoSuchElementException if there are no more tokens in this tokenizer's string.

See also

[java.util.Enumeration](#)
[nextToken\(\)](#)

Since

ostermillerutils 1.00.00

Definition at line 867 of file [StringTokenizer.java](#).

5.9.3.12 String Parser.StringTokenizer.nextToken (String nontokenDelims)

Returns the next token in this string tokenizer's string.

Is equivalent to `nextToken(delims, null)`.

Parameters

nontokenDelims the new set of nontoken delimiters (the set of token delimiters will be empty).

Returns

the next token, after switching to the new delimiter set.

Exceptions

NoSuchElementException if there are no more tokens in this tokenizer's string.

See also

[nextToken\(String,String\)](#)

Since

ostermillerutils 1.00.00

Definition at line 810 of file [StringTokenizer.java](#).

5.9.3.13 String Parser.StringTokenizer.nextToken (String delims, boolean delimsAreTokens)

Returns the next token in this string tokenizer's string.

Is equivalent to:

- If the second parameter is `false` -- `nextToken(delims, null)`
- If the second parameter is `true` -- `nextToken(null, delims)`

Parameters

delims the new set of token or nontoken delimiters.

delimsAreTokens flag indicating whether the first parameter specifies token or nontoken delimiters: `false` -- the first parameter specifies nontoken delimiters, the set of token delimiters is empty; `true` -- the first parameter specifies token delimiters, the set of nontoken delimiters is empty.

Returns

the next token, after switching to the new delimiter set.

Exceptions

NoSuchElementException if there are no more tokens in this tokenizer's string.

See also

[nextToken\(String,String\)](#)

Since

ostermillerutils 1.00.00

Definition at line 792 of file [StringTokenizer.java](#).

5.9.3.14 String Parser.StringTokenizer.nextToken (String *nontokenDelims*, String *tokenDelims*, boolean *returnEmptyTokens*)

Returns the next token in this string tokenizer's string.

First, the sets of token and nontoken delimiters are changed to be the `tokenDelims` and `nontokenDelims`, respectively; and whether or not to return empty tokens is set. Then the next token (with respect to new delimiters) in the string after the current position is returned.

The current position is set after the token returned.

The new delimiter set remains the one used for this call and empty tokens are returned in the future as they are in this call.

Parameters

nontokenDelims the new set of nontoken delimiters.

tokenDelims the new set of token delimiters.

returnEmptyTokens true if empty tokens may be returned; false otherwise.

Returns

the next token, after switching to the new delimiter set.

Exceptions

NoSuchElementException if there are no more tokens in this tokenizer's string.

See also

[nextToken\(\)](#)

Since

ostermillerutils 1.00.00

Definition at line 762 of file [StringTokenizer.java](#).

5.9.3.15 String Parser.StringTokenizer.nextToken (String *nontokenDelims*, String *tokenDelims*)

Returns the next token in this string tokenizer's string.

First, the sets of token and nontoken delimiters are changed to be the `tokenDelims` and `nontokenDelims`, respectively. Then the next token (with respect to new delimiters) in the string after the current position is returned.

The current position is set after the token returned.

The new delimiter sets remains the used ones after this call.

Parameters

nontokenDelims the new set of nontoken delimiters.

tokenDelims the new set of token delimiters.

Returns

the next token, after switching to the new delimiter set.

Exceptions

NoSuchElementException if there are no more tokens in this tokenizer's string.

See also

[nextToken\(\)](#)

Since

ostermillerutils 1.00.00

Definition at line 734 of file [StringTokenizer.java](#).

5.9.3.16 String Parser.StringTokenizer.nextToken ()

Returns the next token from this string tokenizer.

The current position is set after the token returned.

Returns

the next token from this string tokenizer.

Exceptions

NoSuchElementException if there are no more tokens in this tokenizer's string.

Since

ostermillerutils 1.00.00

Definition at line 369 of file [StringTokenizer.java](#).

5.9.3.17 String Parser.StringTokenizer.peek ()

Returns the same value as [nextToken\(\)](#) but does not alter the internal state of the Tokenizer. Subsequent calls to [peek\(\)](#) or a call to [nextToken\(\)](#) will return the same token again.

Returns

the next token from this string tokenizer.

Exceptions

NoSuchElementException if there are no more tokens in this tokenizer's string.

Since

ostermillerutils 1.00.00

Definition at line 997 of file [StringTokenizer.java](#).

5.9.3.18 void Parser.StringTokenizer.remove ()

This implementation always throws *UnsupportedOperationException*. It exists so that this class can implement the `Iterator` interface.

Exceptions

UnsupportedOperationException always is thrown.

See also

`java.util.Iterator`

Since

ostermillerutils 1.00.00

Definition at line 912 of file [StringTokenizer.java](#).

5.9.3.19 String Parser.StringTokenizer.restOfText ()

Retrieves the rest of the text as a single token. After calling this method `hasMoreTokens()` will always return false.

Returns

any part of the text that has not yet been tokenized.

Since

ostermillerutils 1.00.00

Definition at line 982 of file [StringTokenizer.java](#).

5.9.3.20 void Parser.StringTokenizer.setDelimiters (String *nontokenDelims*, String *tokenDelims*, boolean *returnEmptyTokens*)

Set the delimiters used to this set of delimiters.

Parameters

nontokenDelims the new set of nontoken delimiters.

tokenDelims the new set of token delimiters.

returnEmptyTokens true if empty tokens may be returned; false otherwise.

Since

ostermillerutils 1.00.00

Definition at line 526 of file [StringTokenizer.java](#).

5.9.3.21 void Parser.StringTokenizer.setDelimiters (String *nontokenDelims*, String *tokenDelims*)

Set the delimiters used to this set of delimiters.

Parameters

nontokenDelims the new set of nontoken delimiters.

tokenDelims the new set of token delimiters.

Since

ostermillerutils 1.00.00

Definition at line 513 of file [StringTokenizer.java](#).

5.9.3.22 void Parser.StringTokenizer.setDelimiters (String *delims*, boolean *delimsAreTokens*)

Set the delimiters used to this set of delimiters.

Parameters

delims the new set of delimiters.

delimsAreTokens flag indicating whether the first parameter specifies token or nontoken delimiters: false -- the first parameter specifies nontoken delimiters, the set of token delimiters is empty; true -- the first parameter specifies token delimiters, the set of nontoken delimiters is empty.

Since

ostermillerutils 1.00.00

Definition at line 501 of file [StringTokenizer.java](#).

5.9.3.23 void Parser.StringTokenizer.setDelimiters (String *delims*)

Set the delimiters used to this set of (nontoken) delimiters.

Parameters

delims the new set of nontoken delimiters (the set of token delimiters will be empty).

Since

ostermillerutils 1.00.00

Definition at line 486 of file [StringTokenizer.java](#).

5.9.3.24 void Parser.StringTokenizer.setReturnEmptyTokens (boolean *returnEmptyTokens*)

Set whether empty tokens should be returned from this point in in the tokenizing process onward.

Empty tokens occur when two delimiters are next to each other or a delimiter occurs at the beginning or end of a string. If empty tokens are set to be returned, and a comma is the non token delimiter, the following table shows how many tokens are in each string.

String		Number of tokens	
"one,two"		2 - normal case with no empty tokens.	
"one,,three"		3 including the empty token in the middle.	
"one,"		2 including the empty token at the end.	
",two"		2 including the empty token at the beginning.	
","		2 including the empty tokens at the beginning and the ends.	
""		1 - all strings will have at least one token if empty tokens are returned.	

Parameters

returnEmptyTokens true iff empty tokens should be returned.

Since

ostermillerutils 1.00.00

Definition at line 937 of file [StringTokenizer.java](#).

5.9.3.25 void Parser.StringTokenizer.setText (String text)

Set the text to be tokenized in this [StringTokenizer](#).

This is useful when for [StringTokenizer](#) re-use so that new string tokenizers do not have to be created for each string you want to tokenize.

The string will be tokenized from the beginning of the string.

Parameters

text a string to be parsed.

Exceptions

NullPointerException if text is null.

Since

ostermillerutils 1.00.00

Definition at line 255 of file [StringTokenizer.java](#).

5.9.3.26 boolean Parser.StringTokenizer.skipDelimiters ()

Advances the current position so it is before the next token.

This method skips nontoken delimiters but does not skip token delimiters.

This method is useful when switching to the new delimiter sets (see the second example in the class comment.)

Returns

`true` if there are more tokens, `false` otherwise.

Since

ostermillerutils 1.00.00

Definition at line 400 of file [StringTokenizer.java](#).

5.9.3.27 String [] Parser.StringTokenizer.toArray ()

Retrieve all of the remaining tokens in a String array. This method uses the options that are currently set for the tokenizer and will advance the state of the tokenizer such that `hasMoreTokens ()` will return false.

Returns

an array of tokens from this tokenizer.

Since

ostermillerutils 1.00.00

Definition at line 966 of file [StringTokenizer.java](#).

5.9.4 Member Data Documentation**5.9.4.1 int Parser.StringTokenizer.delimsChangedPosition [protected]**

Indicates at which position the delimiters last changed. This will effect how null tokens are returned. Any time that delimiters are changed, the string will be treated as if it is being parsed from position zero, ie, null strings are possible at the very beginning.

Since

ostermillerutils 1.00.00

Definition at line 123 of file [StringTokenizer.java](#).

5.9.4.2 boolean Parser.StringTokenizer.emptyReturned [protected]

One of two variables used to maintain state through the tokenizing process.

true if and only if is found that an empty token should be returned or if empty token was the last thing returned.

If returnEmptyTokens in false, then this variable will always be false.

Since

ostermillerutils 1.00.00

Definition at line 85 of file [StringTokenizer.java](#).

5.9.4.3 char Parser.StringTokenizer.maxDelimChar [protected]

Stores the value of the delimiter character with the highest value. It is used to optimize the detection of delimiter characters. The common case will be that the int values of delimiters will be less than that of most characters in the string (, or space less than any letter for example). Given this, we can check easily check to see if a character is not a delimiter by comparing it to the max delimiter. If it is greater than the max delimiter, then it is no a delimiter otherwise we have to do some more in depth analysis. (ie search the delimiter string.) This will reduce the running time of the algorithm not to depend on the length of the delimiter string for the common case.

Since

ostermillerutils 1.00.00

Definition at line 102 of file [StringTokenizer.java](#).

5.9.4.4 String Parser.StringTokenizer.nontokenDelims [protected]

The set of nontoken delimiters.

Since

ostermillerutils 1.00.00

Definition at line 51 of file [StringTokenizer.java](#).

5.9.4.5 int Parser.StringTokenizer.position [protected]

One of two variables used to maintain state through the tokenizing process.

Represents the position at which we should start looking for the next token (the position of the character immediately following the end of the last token, or 0 to start), or -1 if the entire string has been examined.

Since

ostermillerutils 1.00.00

Definition at line 71 of file [StringTokenizer.java](#).

5.9.4.6 boolean Parser.StringTokenizer.returnEmptyTokens [protected]

Whether empty tokens should be returned. ie if "" should be returned when text starts with a delim, has two delims next to each other, or ends with a delim.

Since

ostermillerutils 1.00.00

Definition at line 112 of file [StringTokenizer.java](#).

5.9.4.7 int Parser.StringTokenizer.strLength [protected]

The length of the text. Cached for performance. This should be set whenever the string we are working with is changed.

Since

ostermillerutils 1.00.00

Definition at line 44 of file [StringTokenizer.java](#).

5.9.4.8 String Parser.StringTokenizer.text [protected]

The string to be tokenized. The code relies on this to never be null.

Since

ostermillerutils 1.00.00

Definition at line 35 of file [StringTokenizer.java](#).

5.9.4.9 int Parser.StringTokenizer.tokenCount [protected]

A cache of the token count. This variable should be -1 if the token have not yet been counted. It should be greater than or equal to zero if the tokens have been counted.

Since

ostermillerutils 1.00.00

Definition at line 132 of file [StringTokenizer.java](#).

5.9.4.10 String Parser.StringTokenizer.tokenDelims [protected]

The set of token delimiters.

Since

ostermillerutils 1.00.00

Definition at line 58 of file [StringTokenizer.java](#).

The documentation for this class was generated from the following file:

- [src/Parser/StringTokenizer.java](#)

5.10 Parser.UnsupportedOperationException Class Reference

5.10.1 Detailed Description

Definition at line 1019 of file [StringTokenizer.java](#).

The documentation for this class was generated from the following file:

- [src/Parser/StringTokenizer.java](#)

5.11 Core.User Class Reference

Public Member Functions

- [User \(\)](#)
- [User \(Integer id, String username, String password, Vector licenseCollection\)](#)
- [Integer getId \(\)](#)
- [void setId \(Integer id\)](#)
- [Vector getLicenseCollection \(\)](#)
- [void setLicenseCollection \(Vector licenseCollection\)](#)
- [String getPassword \(\)](#)
- [void setPassword \(String password\)](#)
- [String getUsername \(\)](#)
- [void setUsername \(String username\)](#)
- [String toString \(\)](#)

5.11.1 Detailed Description

Author

Rodrigo Ramírez Reyes

Definition at line 15 of file [User.java](#).

5.11.2 Constructor & Destructor Documentation

5.11.2.1 Core.User.User ()

Función constructor vacío.

Definition at line 33 of file [User.java](#).

5.11.2.2 Core.User.User (Integer id, String username, String password, Vector licenseCollection)

Constructor del objeto [User](#).

Parameters

id Id del usuario

username Nombre del usuario

password Password del usuario

licenseCollection Arreglo de permisos del usuario

Definition at line 46 of file [User.java](#).

5.11.3 Member Function Documentation

5.11.3.1 Integer Core.User.getId ()

Función para obtener el Id de un usuario

Returns

Id de un usuario

Definition at line 59 of file [User.java](#).

5.11.3.2 Vector Core.User.getLicenseCollection ()

Función para obtener el arreglo de permisos de un usuario

Returns

Arreglo de permisos de un usuario

Definition at line 79 of file [User.java](#).

5.11.3.3 String Core.User.getPassword ()

Función para obtener el Password de un usuario

Returns

Regresa el password de un usuario

Definition at line 99 of file [User.java](#).

5.11.3.4 String Core.User.getUsername ()

Función para obtener el nombre de un usuario

Returns

Regresa el nombre de un usuario

Definition at line 119 of file [User.java](#).

5.11.3.5 void Core.User.setId (Integer *id*)

Función para fijar el Id de un usuario

Parameters

id Id de un usuario de tipo Integer

Definition at line 69 of file [User.java](#).

5.11.3.6 void Core.User.setLicenseCollection (Vector *licenseCollection*)

Función para fijar el arreglo de permisos de un usuario

Parameters

licenseCollection Arreglo de permisos de un usuario de tipo Vector

Definition at line 89 of file [User.java](#).

5.11.3.7 void Core.User.setPassword (String *password*)

Función para fijar el password de un usuario

Parameters

password Password de un usuario de tipo String

Definition at line 109 of file [User.java](#).

5.11.3.8 void Core.User.setUsername (String *username*)

Función para fijar el nombre de un usuario

Parameters

username El nombre de un usuario de tipo String

Definition at line 129 of file [User.java](#).

5.11.3.9 String Core.User.toString ()

Función que sobrescribe al método toString

Returns

Una cadena con todos los parámetros

Definition at line 139 of file [User.java](#).

The documentation for this class was generated from the following file:

- [src/Core/User.java](#)

Chapter 6

File Documentation

6.1 src/Core/File.java File Reference

Classes

- class [Core.File](#)

Packages

- package [Core](#)

6.2 src/Core/File.java

```
00001 /*
00002  * To change this template, choose Tools | Templates
00003  * and open the template in the editor.
00004  */
00005
00006 package Core;
00007
00008 import java.util.Date;
00009 import java.util.Enumeration;
00010 import java.util.Vector;
00011 import javax.microedition.io.Connector;
00012 import javax.microedition.io.file.FileConnection;
00013 import javax.microedition.lcdui.List;
00014
00019 public class File {
00020
00027     private Integer id;
00028     private String filename;
00029     private int filesize;
00030     private String localpath;
00031
00032     private Date creationdate;
00033     private Date modificationdate;
00034     private Vector licenseColletion;
00035
00036
00041     public File() {
00042     }
00043
00044
00056     public File(Integer id, String filename, int filesize, String localpath, Date
creationdate, Date modificationdate, Vector licenseColletion) {
00057         this.id = id;
00058         this.filename = filename;
00059         this.filesize = filesize;
00060         this.localpath = localpath;
00061
00062         this.creationdate = creationdate;
00063         this.modificationdate = modificationdate;
00064         this.licenseColletion = licenseColletion;
00065     }
00066
00073     public Date getCreationdate() {
00074         return creationdate;
00075     }
00076
00083     public void setCreationdate(Date creationdate) {
00084         this.creationdate = creationdate;
00085     }
00086
00093     public String getFilename() {
00094         return filename;
00095     }
00096
00103     public void setFilename(String filename) {
```

```
00104         this.filename = filename;
00105     }
00106
00113     public int getFilesize() {
00114         return filesize;
00115     }
00116
00123     public void setFilesize(int filesize) {
00124         this.filesize = filesize;
00125     }
00126
00133     public Integer getId() {
00134         return id;
00135     }
00136
00143     public void setId(Integer id) {
00144         this.id = id;
00145     }
00146
00153     public Vector getLicenseColletion() {
00154         return licenseColletion;
00155     }
00156
00163     public void setLicenseColletion(Vector licenseColletion) {
00164         this.licenseColletion = licenseColletion;
00165     }
00166
00173     public String getLocalpath() {
00174         return localpath;
00175     }
00176
00183     public void setLocalpath(String localpath) {
00184         this.localpath = localpath;
00185     }
00186
00193     public Date getModificationdate() {
00194         return modificationdate;
00195     }
00196
00203     public void setModificationdate(Date modificationdate) {
00204         this.modificationdate = modificationdate;
00205     }
00206
00207
00213     public String toString()
00214     {
00215         String str = "";
00216         str += id + "\n";
00217         str += filename + "\n";
00218         str += filesize + "\n";
00219         str += localpath + "\n";
00220
00221
00222         return str;
00223     }
00224
00233     public static boolean writeFile(String path, byte[] data)
```

```
00234     {
00235         javax.microedition.io.Connection c = null;
00236         java.io.OutputStream os = null;
00237         System.out.println("Al menos entré");
00238         try {
00239             c = javax.microedition.io.Connector.open("file:///root1/" + path, jav
00240 ax.microedition.io.Connector.READ_WRITE);
00241
00242             javax.microedition.io.file.FileConnection fc =
00243                 (javax.microedition.io.file.FileConnection) c;
00244             if (!fc.exists())
00245                 fc.create();
00246             else
00247                 fc.truncate(0);
00248             os = fc.openOutputStream();
00249             os.write(data);
00250             os.flush();
00251             Enumeration v = javax.microedition.io.file.FileSystemRegistry.listRoo
ts();
00252
00253             //Se mueve hacia un directorio en especifico
00254             //FileConnection currDir = (FileConnection)Connector.open("file:///ro
ot1/");
00255
00256             while(v.hasMoreElements())
00257             {
00258                 System.out.println(v.nextElement() + "\n");
00259             }
00260
00261             System.out.println("Ya estuvo el archivo");
00262             return true;
00263         } catch (Exception e) {
00264             e.printStackTrace();
00265             return false;
00266         } finally {
00267             try {
00268                 if (os != null)
00269                     os.close();
00270                 if (c != null)
00271                     c.close();
00272             } catch (Exception ex) {
00273                 ex.printStackTrace();
00274             }
00275         }
00276     }
00277
00278 }
```

6.3 src/Core/FileContent.java File Reference

Classes

- class [Core.FileContent](#)

Packages

- package [Core](#)

6.4 src/Core/FileContent.java

```
00001 /*
00002  * To change this template, choose Tools | Templates
00003  * and open the template in the editor.
00004  */
00005
00006 package Core;
00007
00012 public class FileContent {
00013
00020     private Integer id;
00021     private String fileid;
00022     private byte[] content;
00023
00029     public FileContent() {
00030     }
00031
00040     public FileContent(Integer id, String fileid, byte[] content) {
00041         this.id = id;
00042         this.fileid = fileid;
00043         this.content = content;
00044     }
00045
00052     public byte[] getContent() {
00053         return content;
00054     }
00055
00062     public void setContent(byte[] content) {
00063         this.content = content;
00064     }
00065
00072     public String getFileid() {
00073         return fileid;
00074     }
00075
00082     public void setFileid(String fileid) {
00083         this.fileid = fileid;
00084     }
00085
00092     public Integer getId() {
00093         return id;
00094     }
00095
00102     public void setId(Integer id) {
00103         this.id = id;
00104     }
00105
00106
00107 }
```


6.5 src/Core/License.java File Reference

Classes

- class [Core.License](#)

Packages

- package [Core](#)

6.6 src/Core/License.java

```
00001 /*
00002  * To change this template, choose Tools | Templates
00003  * and open the template in the editor.
00004  */
00005
00006 package Core;
00007
00012 public class License {
00013
00020     private Integer id;
00021     private String type;
00022     private String userid;
00023     private String fileid;
00024
00030     public License() {
00031     }
00032
00042     public License(Integer id, String type, String userid, String fileid) {
00043         this.id = id;
00044         this.type = type;
00045         this.userid = userid;
00046         this.fileid = fileid;
00047     }
00048
00055     public String getFileid() {
00056         return fileid;
00057     }
00058
00065     public void setFileid(String fileid) {
00066         this.fileid = fileid;
00067     }
00068
00075     public Integer getId() {
00076         return id;
00077     }
00078
00085     public void setId(Integer id) {
00086         this.id = id;
00087     }
00088
00095     public String getType() {
00096         return type;
00097     }
00098
00105     public void setType(String type) {
00106         this.type = type;
00107     }
00108
00115     public String getUserid() {
00116         return userid;
00117     }
00118
00125     public void setUserid(String userid) {
00126         this.userid = userid;
00127     }
}
```

```
00128
00135     public String toString()
00136     {
00137         return id + "\n" + type + "\n" + userid + "\n" + fileid;
00138     }
00139
00140
00141
00142 }
```

6.7 src/Core/Propiedades.java File Reference

Classes

- class [Core.Propiedades](#)

Packages

- package [Core](#)

6.8 src/Core/Propiedades.java

```
00001 /*
00002  * To change this template, choose Tools | Templates
00003  * and open the template in the editor.
00004  */
00005
00006 package Core;
00007
00012 public class Propiedades {
00013
00014
00021     private String urlservidor;
00022
00023
00029     public Propiedades() {
00030
00031     }
00032
00033
00034
00040     public Propiedades(String urlservidor) {
00041         this.urlservidor = urlservidor;
00042     }
00043
00050     public String getUrlservidor() {
00051         return urlservidor;
00052     }
00053
00061     public void setUrlservidor(String urlservidor) {
00062         this.urlservidor = urlservidor;
00063     }
00064
00065
00066
00067
00068 }
```

6.9 src/Core/User.java File Reference

Classes

- class [Core.User](#)

Packages

- package [Core](#)

6.10 src/Core/User.java

```
00001 /*
00002  * To change this template, choose Tools | Templates
00003  * and open the template in the editor.
00004  */
00005
00006 package Core;
00007
00008 import java.util.Vector;
00009 import javax.microedition.amms.control.imageeffect.OverlayControl;
00010
00015 public class User {
00016
00023     private Integer id;
00024     private String username;
00025     private String password;
00026     private Vector licenseCollection;
00027
00033     public User() {
00034         licenseCollection = new Vector();
00035     }
00036
00046     public User(Integer id, String username, String password, Vector licenseColle
00047 tion) {
00048         this.id = id;
00049         this.username = username;
00050         this.password = password;
00051         this.licenseCollection = licenseCollection;
00052     }
00059     public Integer getId() {
00060         return id;
00061     }
00062
00069     public void setId(Integer id) {
00070         this.id = id;
00071     }
00072
00079     public Vector getLicenseCollection() {
00080         return licenseCollection;
00081     }
00082
00089     public void setLicenseCollection(Vector licenseCollection) {
00090         this.licenseCollection = licenseCollection;
00091     }
00092
00099     public String getPassword() {
00100         return password;
00101     }
00102
00109     public void setPassword(String password) {
00110         this.password = password;
00111     }
00112
00119     public String getUsername() {
00120         return username;

```

```
00121     }
00122
00129     public void setUsername(String username) {
00130         this.username = username;
00131     }
00132
00139     public String toString()
00140     {
00141         String str = "";
00142         for(int i = 0; i < licenseColletion.size(); i++)
00143             str += licenseColletion.elementAt(i) + "\n";
00144         return id + "\n" + username + "\n" + password + "\n" + str;
00145     }
00146
00147
00148 }
```


6.11 src/hello/HelloMIDlet.java File Reference

Classes

- class [hello.HelloMIDlet](#)

Packages

- package [hello](#)

6.12 src/hello/HelloMIDlet.java

```
00001 /*
00002  * To change this template, choose Tools | Templates
00003  * and open the template in the editor.
00004  */
00005
00006 package hello;
00007
00008
00009 import Core.File;
00010 import Core.License;
00011 import java.security.NoSuchAlgorithmException;
00012 import javax.microedition.midlet.*;
00013 import javax.microedition.lcdui.*;
00014 import Parser.StringTokenizer;
00015
00016 import javax.microedition.io.Connector;
00017 import javax.microedition.io.HttpConnection;
00018 import java.io.*;
00019 import org.kxml.parser.*;
00020 import org.kxml.kdom.*;
00021 import org.kxml.*;
00022 import org.netbeans.microedition.lcdui.LoginScreen;
00023 import org.netbeans.microedition.lcdui.WaitScreen;
00024 import org.netbeans.microedition.util.SimpleCancellableTask;
00025 import Core.User;
00026 import java.util.Date;
00027 import java.util.Enumeration;
00028 import java.util.Vector;
00029 import javax.microedition.io.file.FileConnection;
00030 import javax.microedition.io.file.FileSystemRegistry;
00031 import org.netbeans.microedition.lcdui.SplashScreen;
00032 import org.netbeans.microedition.lcdui.TableItem;
00033 import org.netbeans.microedition.lcdui.pda.FileBrowser;
00034 import org.netbeans.microedition.svg.SVGMenu;
00035 import security.Sha1;
00036 import java.rmi.*;
00037 import javax.xml.rpc.Stub;
00038
00039
00040 public class HelloMIDlet extends MIDlet implements CommandListener {
00041
00042     private boolean midletPaused = false;
00043
00044
00045
00046
00047
00048
00049
00050     //<editor-fold defaultstate="collapsed" desc=" Generated Fields ">//GEN-BEGIN
00051     :|fields|0|
00052     private Command exitCommand;
00053     private Command okCommand;
00054     private Command okCommand1;
00055     private Command okCommand2;
00056     private Command okCommand4;
00057     private Command okCommand3;
00058     private Command backCommand;
00059     private Command okCommand5;
```

```
00059     private Command okCommand6;
00060     private Command exitCommand1;
00061     private Command backCommand1;
00062     private Command backCommand2;
00063     private Command backCommand3;
00064     private Command okCommand7;
00065     private Command backCommand4;
00066     private Command okCommand8;
00067     private Command backCommand5;
00068     private Command okCommand11;
00069     private Command okCommand10;
00070     private Command backCommand6;
00071     private Command okCommand9;
00072     private Command okCommand13;
00073     private Command okCommand12;
00074     private Command okCommand14;
00075     private Command okCommand15;
00076     private Command okCommand17;
00077     private Command backCommand7;
00078     private Command okCommand16;
00079     private Command backCommand9;
00080     private Command okCommand18;
00081     private Command okCommand19;
00082     private Command backCommand8;
00083     private Command itemCommand;
00084     private Command itemCommand1;
00085     private Command exitCommand2;
00086     private Command cancelCommand;
00087     private Command cancelCommand1;
00088     private Command okCommand20;
00089     private Command itemCommand3;
00090     private Command itemCommand4;
00091     private Command itemCommand2;
00092     private Command okCommand22;
00093     private Command itemCommand5;
00094     private Command okCommand21;
00095     private Command backCommand10;
00096     private Command okCommand24;
00097     private Command cancelCommand2;
00098     private Command okCommand23;
00099     private LoginScreen loginScreen;
00100     private WaitScreen procesoAcceso;
00101     private Alert accesoIncorrecto;
00102     private Alert accesoCorrecto;
00103     private List list;
00104     private FileBrowser fileBrowser;
00105     private List list2;
00106     private List list3;
00107     private WaitScreen waitScreen;
00108     private Alert bitacoraErrorUp;
00109     private Alert bitacoraCreadaUp;
00110     private WaitScreen waitScreen1;
00111     private TextBox textBox1;
00112     private WaitScreen waitScreen2;
00113     private Alert transferenciaErrorUp;
00114     private Alert transferenciaCompletaUp;
00115     private Alert transferenciaErrorDown;
```

```
00116     private Alert transferenciaCompletaDown;
00117     private WaitScreen waitScreen3;
00118     private TextBox textBox5;
00119     private Alert bitacoraErrorDown;
00120     private Alert bitacoraCreadaDown;
00121     private Form form1;
00122     private ChoiceGroup choiceGroup1;
00123     private Form form;
00124     private ChoiceGroup choiceGroup;
00125     private TextBox textBox6;
00126     private Form form3;
00127     private ChoiceGroup choiceGroup2;
00128     private Form form2;
00129     private TextField textField;
00130     private SplashScreen splashScreen;
00131     private WaitScreen waitScreen4;
00132     private Form form5;
00133     private ChoiceGroup choiceGroup4;
00134     private Form form4;
00135     private ChoiceGroup choiceGroup3;
00136     private WaitScreen waitScreen5;
00137     private Alert verArchivosMovil;
00138     private Alert verArchivosServidor;
00139     private WaitScreen waitScreen6;
00140     private TextBox textBox;
00141     private Form form6;
00142     private ChoiceGroup choiceGroup5;
00143     private Alert sincError;
00144     private Alert sincCorrecto;
00145     private SimpleCancellableTask task;
00146     private SimpleCancellableTask task1;
00147     private SimpleCancellableTask task3;
00148     private SimpleCancellableTask task2;
00149     private SimpleCancellableTask task4;
00150     private Image imagen1;
00151     private SimpleCancellableTask task5;
00152     private Image imagen2;
00153     private Image imagen3;
00154     private SimpleCancellableTask task6;
00155     private SimpleCancellableTask task7;
00156     //</editor-fold>//GEN-END:|fields|0|
00157     private Ticker ticker;
00158     private Ticker tickerSalida;
00159     private String currFile;
00160     private TextBox textBoxArchivo;
00161     private TextBox textBoxAlerta;
00162     private TextBox textBoxAlertaCS;
00163     private String xmlarchivolocal;
00164     private boolean seleccion[];
00165     private boolean seleccion1[];
00166     private boolean seleccion2[];
00167     private boolean seleccionSinc[];
00168     private boolean seleccionSinc2[];
00169
00170     private String sel;
00171     private String sel1;
00172     private String sel2;
```

```

00173     private String sel3;
00174     private String sel4;
00175     private String selSincGeneral;
00176     private String selSincReemplazo;
00177
00178     private Vector salidaSeleccion1;
00179     private Vector salidaSeleccion2;
00180     private String usuarioActual;
00181     private String passwordActual;
00182     private String urlDefinida;
00183
00184     private Command cancelCommand25;
00185     private Command okCommand25;
00186
00187     private Command cancelCommand30;
00188     private Command okCommand30;
00189
00190     private boolean espera;
00191     private boolean reemplaza;
00192     private String clienteoservidor;
00193     private String nombreAlerta;
00194
00195
00199     public HelloMIDlet() {
00200     }
00201
00202     //<editor-fold defaultstate="collapsed" desc=" Generated Methods ">//GEN-BEGIN
N:|methods|0|
00203     //</editor-fold>//GEN-END:|methods|0|
00204
00205     //<editor-fold defaultstate="collapsed" desc=" Generated Method: initialize "
>//GEN-BEGIN:|0-initialize|0|0-preInitialize
00210     private void initialize() { //GEN-END:|0-initialize|0|0-preInitialize
00211         // write pre-initialize user code here
00212     //GEN-LINE:|0-initialize|1|0-postInitialize
00213         // write post-initialize user code here
00214     } //GEN-BEGIN:|0-initialize|2|
00215     //</editor-fold>//GEN-END:|0-initialize|2|
00216
00217     //<editor-fold defaultstate="collapsed" desc=" Generated Method: startMIDlet
">//GEN-BEGIN:|3-startMIDlet|0|3-preAction
00221     public void startMIDlet() { //GEN-END:|3-startMIDlet|0|3-preAction
00222         // write pre-action user code here
00223         switchDisplayable(null, getLoginScreen()); //GEN-LINE:|3-startMIDlet|1|3-p
ostAction
00224         // write post-action user code here
00225     } //GEN-BEGIN:|3-startMIDlet|2|
00226     //</editor-fold>//GEN-END:|3-startMIDlet|2|
00227
00228     //<editor-fold defaultstate="collapsed" desc=" Generated Method: resumeMIDlet
">//GEN-BEGIN:|4-resumeMIDlet|0|4-preAction
00232     public void resumeMIDlet() { //GEN-END:|4-resumeMIDlet|0|4-preAction
00233         // write pre-action user code here
00234     //GEN-LINE:|4-resumeMIDlet|1|4-postAction
00235         // write post-action user code here
00236     } //GEN-BEGIN:|4-resumeMIDlet|2|
00237     //</editor-fold>//GEN-END:|4-resumeMIDlet|2|

```



```

00238
00239     //<editor-fold defaultstate="collapsed" desc=" Generated Method: switchDispla
yable ">//GEN-BEGIN:|5-switchDisplayable|0|5-preSwitch
00245     public void switchDisplayable(Alert alert, Displayable nextDisplayable) { //GE
N-END:|5-switchDisplayable|0|5-preSwitch
00246         // write pre-switch user code here
00247         Display display = getDisplay();//GEN-BEGIN:|5-switchDisplayable|1|5-postS
witch
00248         if (alert == null) {
00249             display.setCurrent(nextDisplayable);
00250         } else {
00251             display.setCurrent(alert, nextDisplayable);
00252         } //GEN-END:|5-switchDisplayable|1|5-postSwitch
00253         // write post-switch user code here
00254     } //GEN-BEGIN:|5-switchDisplayable|2|
00255     //</editor-fold>//GEN-END:|5-switchDisplayable|2|
00256
00257     //<editor-fold defaultstate="collapsed" desc=" Generated Method: commandActio
n for Displayables ">//GEN-BEGIN:|7-commandAction|0|7-preCommandAction
00263     public void commandAction(Command command, Displayable displayable) { //GEN-EN
D:|7-commandAction|0|7-preCommandAction
00264         // write pre-action user code here
00265         if (displayable == fileBrowser) { //GEN-BEGIN:|7-commandAction|1|90-preAct
ion
00266             if (command == FileBrowser.SELECT_FILE_COMMAND) { //GEN-END:|7-command
Action|1|90-preAction
00267                 // write pre-action user code here
00268                 //         FileBrowser fil = new FileBrowser(null);
00269                 //         fil.commandAction(command, displayable);
00270                 List curr = (List) displayable;
00271                 currFile = curr.getString(curr.getSelectedIndex());
00272
00273
00274                 //muestraArchivoAlerta(currFile);
00275                 muestraPropiedades(null, currFile);
00276 //GEN-LINE:|7-commandAction|2|90-postAction
00277
00278                 // write post-action user code here
00279             } else if (command == backCommand1) { //GEN-LINE:|7-commandAction|3|11
3-preAction
00280                 // write pre-action user code here
00281                 switchDisplayable(null, getList2());//GEN-LINE:|7-commandAction|4
|113-postAction
00282                 // write post-action user code here
00283             } //GEN-BEGIN:|7-commandAction|5|287-preAction
00284         } else if (displayable == form) {
00285             if (command == backCommand7) { //GEN-END:|7-commandAction|5|287-preAct
ion
00286                 // write pre-action user code here
00287                 switchDisplayable(null, getList());//GEN-LINE:|7-commandAction|6|
287-postAction
00288                 // write post-action user code here
00289             } else if (command == okCommand16) { //GEN-LINE:|7-commandAction|7|268
-preAction
00290                 // write pre-action user code here
00291
00292

```

```

00293
00294         switchDisplayable(null, getWaitScreen2()); //GEN-LINE:|7-commandAc
tion|8|268-postAction
00295
00296         //Muestra las opciones de sincronización
00297
00298         seleccion = new boolean[choiceGroup.size()];
00299         sel = new String("");
00300         choiceGroup.getSelectedFlags(seleccion);
00301         for (int x = 0; x < seleccion.length; x++) {
00302             if (seleccion[x]) {
00303                 sel += choiceGroup.getString(x);
00304                 //sel += "\n";
00305             }
00306         }
00307
00308         //asigna lo seleccionado de la opción de sincronización de
00309         //descarga a la variable selSincGeneral para utilizarla en el proce
so
00310         //de sincronización.
00311         selSincGeneral = sel;
00312
00313         if(!sel.equals("")){
00314             Alert elegido = new Alert("Selección",
00315                 "Elegiste las opciones: " + sel,
00316                 null, AlertType.WARNING);
00317             elegido.setTimeout(Alert.FOREVER);
00318             Display.getDisplay(this).setCurrent(elegido);
00319         }
00320         else{
00321             Alert elegido = new Alert("Selección",
00322                 "Error. No seleccionaste nada.",
00323                 null, AlertType.WARNING);
00324             elegido.setTimeout(Alert.FOREVER);
00325             Display.getDisplay(this).setCurrent(elegido);
00326         }
00327
00328         // write post-action user code here
00329     } //GEN-BEGIN:|7-commandAction|9|290-preAction
00330     } else if (displayable == form1) {
00331         if (command == backCommand8) { //GEN-END:|7-commandAction|9|290-preAct
ion
00332             // write pre-action user code here
00333             switchDisplayable(null, getList()); //GEN-LINE:|7-commandAction|10
|290-postAction
00334             // write post-action user code here
00335         } else if (command == okCommand17) { //GEN-LINE:|7-commandAction|11|27
6-preAction
00336             // write pre-action user code here
00337             switchDisplayable(null, getWaitScreen()); //GEN-LINE:|7-commandAct
ion|12|276-postAction
00338
00339             seleccion1 = new boolean[choiceGroup1.size()];
00340             //StringItem mensaje[] = new StringItem[choiceGroup.size()];
00341             sel1 = new String("");
00342             choiceGroup1.getSelectedFlags(seleccion1);
00343             for (int x = 0; x < seleccion1.length; x++) {

```

```

00344         if (seleccion1[x]) {
00345             sell += choiceGroup1.getString(x);
00346             //sell += "\n";
00347         }
00348     }
00349
00350     //asigna lo seleccionado de la opción de sincronización de
00351     //descarga a la variable selSincGeneral para utilizarla en el proce
so
00352     //de sincronización.
00353     selSincGeneral = sell;
00354
00355     if(!sell.equals("")){
00356         Alert elegido1 = new Alert("Seleccion",
00357             "Elegiste las opciones: " + sell,
00358             null, AlertType.WARNING);
00359         Display.getDisplay(this).setCurrent(elegido1);
00360     }
00361     else{
00362         Alert elegido = new Alert("Seleccion",
00363             "Error. No seleccionaste nada.",
00364             null, AlertType.WARNING);
00365         Display.getDisplay(this).setCurrent(elegido);
00366     }
00367
00368
00369     // write post-action user code here
00370     }//GEN-BEGIN:|7-commandAction|13|306-preAction
00371 } else if (displayable == form2) {
00372     if (command == backCommand9) { //GEN-END:|7-commandAction|13|306-preAc
tion
00373         // write pre-action user code here
00374         switchDisplayable(null, getLoginScreen()); //GEN-LINE:|7-commandAc
tion|14|306-postAction
00375         // write post-action user code here
00376     } else if (command == itemCommand1) { //GEN-LINE:|7-commandAction|15|3
04-preAction
00377         // write pre-action user code here
00378         switchDisplayable(null, getTextBox6()); //GEN-LINE:|7-commandActio
n|16|304-postAction
00379         System.out.println("la cajita 2 dice: " + textField.getString().t
oString());
00380         Parser.RestParser.setDocumentLocal(textField.getString().toString
());
00381         // write post-action user code here
00382     } //GEN-BEGIN:|7-commandAction|17|372-preAction
00383 } else if (displayable == form3) {
00384     if (command == itemCommand2) { //GEN-END:|7-commandAction|17|372-preAc
tion
00385         // write pre-action user code here
00386         boolean[] f = new boolean[choiceGroup2.size()];
00387         for(int in = 0; in < choiceGroup2.size(); in++){
00388             f[in]=true;
00389         }
00390
00391         choiceGroup2.setSelectedFlags(f);
00392     } //GEN-LINE:|7-commandAction|18|372-postAction

```



```

00393         // write post-action user code here
00394     } else if (command == itemCommand3) { //GEN-LINE:|7-commandAction|19|3
74-preAction
00395         // write pre-action user code here
00396         boolean[] f = new boolean[choiceGroup2.size()];
00397         for(int in = 0; in < choiceGroup2.size(); in++){
00398             f[in]=false;
00399         }
00400
00401         choiceGroup2.setSelectedFlags(f);
00402 //GEN-LINE:|7-commandAction|20|374-postAction
00403         // write post-action user code here
00404     } else if (command == okCommand19) { //GEN-LINE:|7-commandAction|21|31
8-preAction
00405         // write pre-action user code here
00406         switchDisplayable(null, getWaitScreen4()); //GEN-LINE:|7-commandAc
tion|22|318-postAction
00407         // write post-action user code here
00408
00409         //Muestra las opciones de sincronización
00410         Vector g = new Vector();
00411         salidaSeleccion1.removeAllElements();
00412
00413         seleccion2 = new boolean[choiceGroup2.size()];
00414         sel2 = new String("");
00415         choiceGroup2.getSelectedFlags(seleccion2);
00416         for (int x = 0; x < seleccion2.length; x++) {
00417             if (seleccion2[x]) {
00418                 salidaSeleccion1.addElement (choiceGroup2.getString(x).toStrin
g());
00419                 sel2 += choiceGroup2.getString(x);
00420                 sel2 += "\n";
00421             }
00422         }
00423
00424         if(!sel2.equals("")){
00425             Alert elegido2 = new Alert("Seleccion",
00426                 "Seleccionaste los siguientes archivos:\n" + sel2,
00427                 null, AlertType.WARNING);
00428             elegido2.setTimeout(Alert.FOREVER);
00429             Display.getDisplay(this).setCurrent (elegido2);
00430         }
00431         else{
00432             Alert elegido2 = new Alert("Seleccion",
00433                 "No seleccionaste ninguna opción de sincronización\n",
00434                 null, AlertType.WARNING);
00435             elegido2.setTimeout(Alert.FOREVER);
00436             Display.getDisplay(this).setCurrent (elegido2);
00437         }
00438
00439
00440         } //GEN-BEGIN:|7-commandAction|23|376-preAction
00441     } else if (displayable == form4) {
00442         if (command == itemCommand4) { //GEN-END:|7-commandAction|23|376-preAc
tion
00443             // write pre-action user code here
00444

```

```

00445         boolean[] j = new boolean[choiceGroup3.size()];
00446         for(int in = 0; in < choiceGroup3.size(); in++){
00447             j[in]=true;
00448         }
00449
00450         choiceGroup3.setSelectedFlags(j);
00451
00452 //GEN-LINE:|7-commandAction|24|376-postAction
00453         // write post-action user code here
00454     } else if (command == itemCommand5) { //GEN-LINE:|7-commandAction|25|3
78-preAction
00455         // write pre-action user code here
00456
00457         boolean[] h = new boolean[choiceGroup3.size()];
00458         for(int in = 0; in < choiceGroup3.size(); in++){
00459             h[in]=false;
00460         }
00461
00462         choiceGroup3.setSelectedFlags(h);
00463
00464 //GEN-LINE:|7-commandAction|26|378-postAction
00465         // write post-action user code here
00466     } else if (command == okCommand21) { //GEN-LINE:|7-commandAction|27|35
9-preAction
00467         // write pre-action user code here
00468         switchDisplayable(null, getWaitScreen5()); //GEN-LINE:|7-commandAc
tion|28|359-postAction
00469         // write post-action user code here
00470
00471         //Muestra las opciones de sincronización
00472         salidaSeleccion2.removeAllElements();
00473
00474         seleccion2 = new boolean[choiceGroup3.size()];
00475         sel2 = new String("");
00476         choiceGroup3.getSelectedFlags(seleccion2);
00477         for (int x = 0; x < seleccion2.length; x++) {
00478             if (seleccion2[x]) {
00479                 salidaSeleccion2.addElement(choiceGroup3.getString(x).toStrin
g());
00480
00481                 sel2 += choiceGroup3.getString(x);
00482                 sel2 += "\n";
00483             }
00484         }
00485         if(!sel2.equals("")){
00486             Alert elegido3 = new Alert("Seleccion",
00487                 "Seleccionaste los siguientes archivos:\n" + sel2,
00488                 null, AlertType.WARNING);
00489             elegido3.setTimeout(Alert.FOREVER);
00490             Display.getDisplay(this).setCurrent(elegido3);
00491         }
00492         else{
00493             Alert elegido3 = new Alert("Seleccion",
00494                 "No seleccionaste ninguna opción de sincronización\n",
00495                 null, AlertType.WARNING);
00496             elegido3.setTimeout(Alert.FOREVER);
00497             Display.getDisplay(this).setCurrent(elegido3);

```

```

00498         }
00499
00500         } //GEN-BEGIN:|7-commandAction|29|386-preAction
00501     } else if (displayable == form5) {
00502         if (command == backCommand10) { //GEN-END:|7-commandAction|29|386-preA
ction
00503             // write pre-action user code here
00504
00505             switchDisplayable(null, getList()); //GEN-LINE:|7-commandAction|30
|386-postAction
00506             // write post-action user code here
00507
00508         } else if (command == okCommand22) { //GEN-LINE:|7-commandAction|31|38
4-preAction
00509             // write pre-action user code here
00510             switchDisplayable(null, getTextBox()); //GEN-LINE:|7-commandAction
|32|384-postAction
00511
00512             seleccionSinc = new boolean[choiceGroup4.size()];
00513             //StringItem mensaje[] = new StringItem[choiceGroup.size()];
00514             sel3 = new String("");
00515             choiceGroup4.getSelectedFlags(seleccionSinc);
00516             for (int x = 0; x < seleccionSinc.length; x++) {
00517                 if (seleccionSinc[x]) {
00518                     sel3 += choiceGroup4.getString(x);
00519                     //sel1 += "\n";
00520                 }
00521             }
00522
00523             //asigna lo seleccionado de la opción de sincronización de
00524             //descarga a la variable selSincGeneral para utilizarla en el proce
so
00525             //de sincronización.
00526             selSincGeneral = sel3;
00527             System.out.println("Seleccionaste :" + selSincGeneral);
00528
00529             if (!sel3.equals("")) {
00530                 Alert elegidoSinc = new Alert("Seleccion",
00531                     "Elegiste la opción: " + sel3,
00532                     null, AlertType.WARNING);
00533                 elegidoSinc.setTimeout(2000);
00534                 Display.getDisplay(this).setCurrent(elegidoSinc);
00535             }
00536             else {
00537                 Alert elegidoSinc = new Alert("Seleccion",
00538                     "Error. No seleccionaste nada.",
00539                     null, AlertType.WARNING);
00540                 elegidoSinc.setTimeout(2000);
00541                 Display.getDisplay(this).setCurrent(elegidoSinc);
00542             }
00543
00544             // write post-action user code here
00545         } //GEN-BEGIN:|7-commandAction|33|420-preAction
00546     } else if (displayable == form6) {
00547         if (command == okCommand23) { //GEN-END:|7-commandAction|33|420-preAct
ion
00548             // write pre-action user code here

```



```

00549
00550         switchDisplayable(null, getWaitScreen6()); //GEN-LINE:|7-commandAc
tion|34|420-postAction
00551         // write post-action user code here
00552
00553         seleccionSinc2 = new boolean[choiceGroup5.size()];
00554         //StringItem mensaje[] = new StringItem[choiceGroup.size()];
00555         sel4 = new String("");
00556         choiceGroup5.getSelectedFlags(seleccionSinc2);
00557         for (int x = 0; x < seleccionSinc2.length; x++) {
00558             if (seleccionSinc2[x]) {
00559                 sel4 += choiceGroup5.getString(x);
00560             }
00561         }
00562
00563         //asigna lo seleccionado de la opción de sincronización de
00564         //descarga a la variable selSincGeneral para utilizarla en el proce
so
00565         //de sincronización.
00566         selSincReemplazo = sel4;
00567         System.out.println("Seleccionaste :" + selSincReemplazo);
00568
00569
00570         } //GEN-BEGIN:|7-commandAction|35|66-preAction
00571     } else if (displayable == list) {
00572         if (command == List.SELECT_COMMAND) { //GEN-END:|7-commandAction|35|66
-preAction
00573             // write pre-action user code here
00574             listAction(); //GEN-LINE:|7-commandAction|36|66-postAction
00575             // write post-action user code here
00576             } //GEN-BEGIN:|7-commandAction|37|118-preAction
00577         } else if (displayable == list2) {
00578             if (command == List.SELECT_COMMAND) { //GEN-END:|7-commandAction|37|11
8-preAction
00579                 // write pre-action user code here
00580                 list2Action(); //GEN-LINE:|7-commandAction|38|118-postAction
00581                 // write post-action user code here
00582             } else if (command == backCommand3) { //GEN-LINE:|7-commandAction|39|1
133-preAction
00583                 // write pre-action user code here
00584                 switchDisplayable(null, getList()); //GEN-LINE:|7-commandAction|40
|133-postAction
00585                 // write post-action user code here
00586                 } //GEN-BEGIN:|7-commandAction|41|125-preAction
00587             } else if (displayable == list3) {
00588                 if (command == List.SELECT_COMMAND) { //GEN-END:|7-commandAction|41|12
5-preAction
00589                     // write pre-action user code here
00590                     list3Action(); //GEN-LINE:|7-commandAction|42|125-postAction
00591                     // write post-action user code here
00592                 } else if (command == backCommand2) { //GEN-LINE:|7-commandAction|43|1
29-preAction
00593                     // write pre-action user code here
00594                     switchDisplayable(null, getList2()); //GEN-LINE:|7-commandAction|4
4|129-postAction
00595                     // write post-action user code here
00596                 } //GEN-BEGIN:|7-commandAction|45|24-preAction

```

```

00597         } else if (displayable == loginScreen) {
00598             if (command == LoginScreen.LOGIN_COMMAND) { //GEN-END:|7-commandAction
|45|24-preAction
00599                 // write pre-action user code here
00600                 switchDisplayable(null, getProcesoAcceso()); //GEN-LINE:|7-command
Action|46|24-postAction
00601                 // write post-action user code here
00602             } else if (command == exitCommand1) { //GEN-LINE:|7-commandAction|47|1
16-preAction
00603                 // write pre-action user code here
00604                 exitMIDlet();
00605 //GEN-LINE:|7-commandAction|48|116-postAction
00606                 // write post-action user code here
00607             } else if (command == itemCommand) { //GEN-LINE:|7-commandAction|49|29
5-preAction
00608                 // write pre-action user code here
00609                 switchDisplayable(null, getForm2()); //GEN-LINE:|7-commandAction|5
0|295-postAction
00610                 // write post-action user code here
00611             } //GEN-BEGIN:|7-commandAction|51|30-preAction
00612         } else if (displayable == procesoAcceso) {
00613             if (command == WaitScreen.FAILURE_COMMAND) { //GEN-END:|7-commandActio
n|51|30-preAction
00614                 // write pre-action user code here
00615                 switchDisplayable(getAccesoIncorrecto(), getLoginScreen()); //GEN-
LINE:|7-commandAction|52|30-postAction
00616                 // write post-action user code here
00617             } else if (command == WaitScreen.SUCCESS_COMMAND) { //GEN-LINE:|7-comm
andAction|53|29-preAction
00618                 // write pre-action user code here
00619                 switchDisplayable(getAccesoCorrecto(), getList()); //GEN-LINE:|7-c
ommandAction|54|29-postAction
00620                 // write post-action user code here
00621             } //GEN-BEGIN:|7-commandAction|55|341-preAction
00622         } else if (displayable == splashScreen) {
00623             if (command == SplashScreen.DISMISS_COMMAND) { //GEN-END:|7-commandAct
ion|55|341-preAction
00624                 // write pre-action user code here
00625                 exitMIDlet(); //GEN-LINE:|7-commandAction|56|341-postAction
00626                 // write post-action user code here
00627             } //GEN-BEGIN:|7-commandAction|57|446-preAction
00628         } else if (displayable == textBox) {
00629             if (command == cancelCommand2) { //GEN-END:|7-commandAction|57|446-pre
Action
00630                 // write pre-action user code here
00631                 switchDisplayable(null, getList()); //GEN-LINE:|7-commandAction|58
|446-postAction
00632                 // write post-action user code here
00633             } else if (command == okCommand24) { //GEN-LINE:|7-commandAction|59|44
3-preAction
00634                 // write pre-action user code here
00635                 method(); //GEN-LINE:|7-commandAction|60|443-postAction
00636                 // write post-action user code here
00637             } //GEN-BEGIN:|7-commandAction|61|350-preAction
00638         } else if (displayable == textBox1) {
00639             if (command == cancelCommand1) { //GEN-END:|7-commandAction|61|350-pre
Action

```

```

00640         // write pre-action user code here
00641         switchDisplayable(null, getList());//GEN-LINE:|7-commandAction|62
|350-postAction
00642         // write post-action user code here
00643     } else if (command == okCommand13) {//GEN-LINE:|7-commandAction|63|22
0-preAction
00644         // write pre-action user code here
00645         switchDisplayable(null, getWaitScreen1());//GEN-LINE:|7-commandAc
tion|64|220-postAction
00646         // write post-action user code here
00647     }//GEN-BEGIN:|7-commandAction|65|338-preAction
00648     } else if (displayable == textBox5) {
00649         if (command == cancelCommand) {//GEN-END:|7-commandAction|65|338-preA
ction
00650         // write pre-action user code here
00651         switchDisplayable(null, getList());//GEN-LINE:|7-commandAction|66
|338-postAction
00652         // write post-action user code here
00653     } else if (command == okCommand15) {//GEN-LINE:|7-commandAction|67|24
7-preAction
00654         // write pre-action user code here
00655         switchDisplayable(null, getWaitScreen3());//GEN-LINE:|7-commandAc
tion|68|247-postAction
00656         // write post-action user code here
00657     }//GEN-BEGIN:|7-commandAction|69|313-preAction
00658     } else if (displayable == textBox6) {
00659         if (command == okCommand18) {//GEN-END:|7-commandAction|69|313-preAct
ion
00660         // write pre-action user code here
00661         switchDisplayable(null, getLoginScreen());//GEN-LINE:|7-commandAc
tion|70|313-postAction
00662         // write post-action user code here
00663     }//GEN-BEGIN:|7-commandAction|71|154-preAction
00664     } else if (displayable == waitScreen) {
00665         if (command == WaitScreen.FAILURE_COMMAND) {//GEN-END:|7-commandActio
n|71|154-preAction
00666         // write pre-action user code here
00667         switchDisplayable(getBitacoraErrorUp(), getList());//GEN-LINE:|7-
commandAction|72|154-postAction
00668         // write post-action user code here
00669     } else if (command == WaitScreen.SUCCESS_COMMAND) {//GEN-LINE:|7-comm
andAction|73|153-preAction
00670         // write pre-action user code here
00671         switchDisplayable(getBitacoraCreadaUp(), getTextBox1());//GEN-LIN
E:|7-commandAction|74|153-postAction
00672         // write post-action user code here
00673     }//GEN-BEGIN:|7-commandAction|75|225-preAction
00674     } else if (displayable == waitScreen1) {
00675         if (command == WaitScreen.FAILURE_COMMAND) {//GEN-END:|7-commandActio
n|75|225-preAction
00676         // write pre-action user code here
00677         switchDisplayable(getTransferenciaErrorUp(), getList());//GEN-LIN
E:|7-commandAction|76|225-postAction
00678         // write post-action user code here
00679     } else if (command == WaitScreen.SUCCESS_COMMAND) {//GEN-LINE:|7-comm
andAction|77|224-preAction
00680         // write pre-action user code here

```



```

00681         switchDisplayable(null, getForm4()); //GEN-LINE:|7-commandAction|7
00682     8|224-postAction
00683         // write post-action user code here
00684     } //GEN-BEGIN:|7-commandAction|79|238-preAction
00685     } else if (displayable == waitScreen2) {
00686         if (command == WaitScreen.FAILURE_COMMAND) { //GEN-END:|7-commandActio
00687     n|79|238-preAction
00688         // write pre-action user code here
00689     switchDisplayable(getBitacoraErrorDown(), getList()); //GEN-LINE:|
00690     7-commandAction|80|238-postAction
00691         // write post-action user code here
00692     } else if (command == WaitScreen.SUCCESS_COMMAND) { //GEN-LINE:|7-comm
00693     andAction|81|237-preAction
00694         // write pre-action user code here
00695     switchDisplayable(getBitacoraCreadaDown(), getTextBox5()); //GEN-L
00696     INE:|7-commandAction|82|237-postAction
00697         // write post-action user code here
00698     } //GEN-BEGIN:|7-commandAction|83|250-preAction
00699     } else if (displayable == waitScreen3) {
00700         if (command == WaitScreen.FAILURE_COMMAND) { //GEN-END:|7-commandActio
00701     n|83|250-preAction
00702         // write pre-action user code here
00703     switchDisplayable(getTransferenciaErrorDown(), getList()); //GEN-L
00704     INE:|7-commandAction|84|250-postAction
00705         // write post-action user code here
00706     } else if (command == WaitScreen.SUCCESS_COMMAND) { //GEN-LINE:|7-comm
00707     andAction|85|249-preAction
00708         // write pre-action user code here
00709     switchDisplayable(null, getForm3()); //GEN-LINE:|7-commandAction|8
00710     6|249-postAction
00711         // write post-action user code here
00712     } //GEN-BEGIN:|7-commandAction|87|329-preAction
00713     } else if (displayable == waitScreen4) {
00714         if (command == WaitScreen.FAILURE_COMMAND) { //GEN-END:|7-commandActio
00715     n|87|329-preAction
00716         // write pre-action user code here
00717     switchDisplayable(getTransferenciaErrorDown(), getList()); //GEN-L
00718     INE:|7-commandAction|88|329-postAction
00719         // write post-action user code here
00720     } else if (command == WaitScreen.SUCCESS_COMMAND) { //GEN-LINE:|7-comm
00721     andAction|89|328-preAction
00722         // write pre-action user code here
00723     switchDisplayable(getTransferenciaCompletaDown(), getList()); //GE
00724     N-LINE:|7-commandAction|90|328-postAction
00725         // write post-action user code here
00726     } //GEN-BEGIN:|7-commandAction|91|356-preAction
00727     } else if (displayable == waitScreen5) {
00728         if (command == WaitScreen.FAILURE_COMMAND) { //GEN-END:|7-commandActio
00729     n|91|356-preAction
00730         // write pre-action user code here
00731     switchDisplayable(getTransferenciaErrorUp(), getList()); //GEN-LIN
00732     E:|7-commandAction|92|356-postAction
00733         // write post-action user code here
00734     } else if (command == WaitScreen.SUCCESS_COMMAND) { //GEN-LINE:|7-comm
00735     andAction|93|355-preAction
00736         // write pre-action user code here
00737     switchDisplayable(getTransferenciaCompletaUp(), getList()); //GEN-

```



```

LINE:|7-commandAction|94|355-postAction
00722         // write post-action user code here
00723     } //GEN-BEGIN:|7-commandAction|95|412-preAction
00724     } else if (displayable == waitScreen6) {
00725         if (command == WaitScreen.FAILURE_COMMAND) { //GEN-END:|7-commandActio
n|95|412-preAction
00726         // write pre-action user code here
00727         switchDisplayable(getSincError(), getList()); //GEN-LINE:|7-comman
dAction|96|412-postAction
00728         // write post-action user code here
00729     } else if (command == WaitScreen.SUCCESS_COMMAND) { //GEN-LINE:|7-comm
andAction|97|411-preAction
00730         // write pre-action user code here
00731         switchDisplayable(getSincCorrecto(), getList()); //GEN-LINE:|7-com
mandAction|98|411-postAction
00732         // write post-action user code here
00733     } //GEN-BEGIN:|7-commandAction|99|7-postCommandAction
00734 } //GEN-END:|7-commandAction|99|7-postCommandAction
00735 // write post-action user code here
00736 else if (displayable == textBoxAlerta) {
00737     if (command == okCommand25) {
00738         // write pre-action user code here
00739         espera = false;
00740         reemplaza = true;
00741         // write post-action user code here
00742     }
00743     else if (command == cancelCommand25) {
00744         // write pre-action user code here
00745         espera = false;
00746         // write post-action user code here
00747     }
00748 }
00749 else if (displayable == textBoxAlertaCS) {
00750     if (command == okCommand30) {
00751         // write pre-action user code here
00752         //clienteoservidor = "servidor";
00753         espera = false;
00754
00755         //reemplaza = true;
00756         // write post-action user code here
00757     }
00758     else if (command == cancelCommand30) {
00759         // write pre-action user code here
00760         clienteoservidor = "cliente";
00761         espera = false;
00762
00763         // write post-action user code here
00764     }
00765 }
00766 } //GEN-BEGIN:|7-commandAction|100|411-postAction
00767 //</editor-fold> //GEN-END:|7-commandAction|100|411-postAction
00768
00769
00770
00771
00772
00773

```

```

00774
00775
00776
00777
00778
00779
00780
00781
00782
00783
00784
00785
00786
00787
00788
00789    //<editor-fold defaultstate="collapsed" desc=" Generated Getter: exitCommand
">//GEN-BEGIN:|18-getter|0|18-preInit
00794    public Command getExitCommand() {
00795        if (exitCommand == null) { //GEN-END:|18-getter|0|18-preInit
00796            // write pre-init user code here
00797            exitCommand = new Command("Exit", Command.EXIT, 0); //GEN-LINE:|18-get
ter|1|18-postInit
00798            // write post-init user code here
00799            exitCommand = new Command("Regresar", Command.EXIT, 0);
00800        } //GEN-BEGIN:|18-getter|2|
00801        return exitCommand;
00802    }
00803    //</editor-fold>//GEN-END:|18-getter|2|
00804
00805    //<editor-fold defaultstate="collapsed" desc=" Generated Getter: loginScreen
">//GEN-BEGIN:|22-getter|0|22-preInit
00810    public LoginScreen getLoginScreen() {
00811        if (loginScreen == null) { //GEN-END:|22-getter|0|22-preInit
00812            // write pre-init user code here
00813            loginScreen = new LoginScreen(getDisplay()); //GEN-BEGIN:|22-getter|1|
22-postInit
00814            loginScreen.setLabelTexts("Usuario", "Password");
00815            loginScreen.setTitle("Aplicaci\u00F3n m\u00F3vil para la Sincronizaci
\u00F3n de Archivos");
00816            loginScreen.addCommand(LoginScreen.LOGIN_COMMAND);
00817            loginScreen.addCommand(getExitCommand());
00818            loginScreen.addCommand(getItemCommand());
00819            loginScreen.setCommandListener(this);
00820            loginScreen.setBGColor(-16737793);
00821            loginScreen.setFGColor(0);
00822            loginScreen.setPassword("ramirez");
00823            loginScreen.setUsername("rodrigo");
00824            loginScreen.setUseLoginButton(false);
00825            loginScreen.setLoginButtonText("Usuario"); //GEN-END:|22-getter|1|22-p
ostInit
00826            // write post-init user code here
00827            loginScreen.setTicker(getTicker());
00828            loginScreen.addCommand(getExitCommand());
00829        } //GEN-BEGIN:|22-getter|2|
00830        return loginScreen;
00831    }
00832    //</editor-fold>//GEN-END:|22-getter|2|

```

```

00833
00834     //<editor-fold defaultstate="collapsed" desc=" Generated Getter: procesoAcces
o ">//GEN-BEGIN:|26-getter|0|26-preInit
00839     public WaitScreen getProcesoAcceso() {
00840         if (procesoAcceso == null) { //GEN-END:|26-getter|0|26-preInit
00841             // write pre-init user code here
00842             procesoAcceso = new WaitScreen(getDisplay()); //GEN-BEGIN:|26-getter|1
|26-postInit
00843             procesoAcceso.setTitle("waitScreen");
00844             procesoAcceso.setCommandListener(this);
00845             procesoAcceso.setTask(getTask()); //GEN-END:|26-getter|1|26-postInit
00846             // write post-init user code here
00847         } //GEN-BEGIN:|26-getter|2|
00848         return procesoAcceso;
00849     }
00850     //</editor-fold>//GEN-END:|26-getter|2|
00851
00852     //<editor-fold defaultstate="collapsed" desc=" Generated Getter: accesoIncorr
ecto ">//GEN-BEGIN:|32-getter|0|32-preInit
00857     public Alert getAccesoIncorrecto() {
00858         if (accesoIncorrecto == null) { //GEN-END:|32-getter|0|32-preInit
00859             // write pre-init user code here
00860             accesoIncorrecto = new Alert("nologin", "El usuario o la contrase\u00
Fla son incorrectas", getImage3(), null); //GEN-BEGIN:|32-getter|1|32-postInit
00861             accesoIncorrecto.setTimeout(2000); //GEN-END:|32-getter|1|32-postInit
00862             // write post-init user code here
00863             //accesoIncorrecto = new Alert("nologin", "El usuario o la contrase\u00
son incorrectas", null, null);
00864         } //GEN-BEGIN:|32-getter|2|
00865         return accesoIncorrecto;
00866     }
00867     //</editor-fold>//GEN-END:|32-getter|2|
00868
00869     //<editor-fold defaultstate="collapsed" desc=" Generated Getter: accesoCorrec
to ">//GEN-BEGIN:|33-getter|0|33-preInit
00874     public Alert getAccesoCorrecto() {
00875         if (accesoCorrecto == null) { //GEN-END:|33-getter|0|33-preInit
00876             // write pre-init user code here
00877             accesoCorrecto = new Alert("login", "Has ingresado correctamente",
getImage2(), null); //GEN-BEGIN:|33-getter|1|33-postInit
00878             accesoCorrecto.setTimeout(2000); //GEN-END:|33-getter|1|33-postInit
00879             // write post-init user code here
00880             //accesoCorrecto = new Alert("login", "Has ingresado correctamente",
null, null);
00881         } //GEN-BEGIN:|33-getter|2|
00882         return accesoCorrecto;
00883     }
00884     //</editor-fold>//GEN-END:|33-getter|2|
00885
00886     //<editor-fold defaultstate="collapsed" desc=" Generated Getter: task ">//GEN
-BEGIN:|31-getter|0|31-preInit
00891     public SimpleCancellableTask getTask() {
00892         if (task == null) { //GEN-END:|31-getter|0|31-preInit
00893             // write pre-init user code here
00894             task = new SimpleCancellableTask(); //GEN-BEGIN:|31-getter|1|31-execut
e
00895         task.setExecutable(new org.netbeans.microedition.util.Executable() {

```



```

00896         public void execute() throws Exception { //GEN-END:|31-getter|1|31
-execute
00897             // write task-execution user code here
00898             login();
00899             } //GEN-BEGIN:|31-getter|2|31-postInit
00900         }); //GEN-END:|31-getter|2|31-postInit
00901         // write post-init user code here
00902         } //GEN-BEGIN:|31-getter|3|
00903         return task;
00904     }
00905     //</editor-fold> //GEN-END:|31-getter|3|
00906
00907
00908
00909     //<editor-fold defaultstate="collapsed" desc=" Generated Getter: okCommand ">
>//GEN-BEGIN:|60-getter|0|60-preInit
00914     public Command getOkCommand() {
00915         if (okCommand == null) { //GEN-END:|60-getter|0|60-preInit
00916             // write pre-init user code here
00917             okCommand = new Command("Ok", Command.OK, 0); //GEN-LINE:|60-getter|1|
60-postInit
00918             // write post-init user code here
00919         } //GEN-BEGIN:|60-getter|2|
00920         return okCommand;
00921     }
00922     //</editor-fold> //GEN-END:|60-getter|2|
00923
00924     //<editor-fold defaultstate="collapsed" desc=" Generated Getter: okCommand1 ">
>//GEN-BEGIN:|68-getter|0|68-preInit
00929     public Command getOkCommand1() {
00930         if (okCommand1 == null) { //GEN-END:|68-getter|0|68-preInit
00931             // write pre-init user code here
00932             okCommand1 = new Command("Ok", Command.OK, 0); //GEN-LINE:|68-getter|1|
|68-postInit
00933             // write post-init user code here
00934         } //GEN-BEGIN:|68-getter|2|
00935         return okCommand1;
00936     }
00937     //</editor-fold> //GEN-END:|68-getter|2|
00938
00939     //<editor-fold defaultstate="collapsed" desc=" Generated Getter: list "> //GEN
-BEGIN:|64-getter|0|64-preInit
00944     public List getList() {
00945         if (list == null) { //GEN-END:|64-getter|0|64-preInit
00946             // write pre-init user code here
00947             list = new List("Men\u00C3FA principal", Choice.IMPLICIT); //GEN-BEGIN:|
64-getter|1|64-postInit
00948             list.append("Descargar archivos", null);
00949             list.append("Subir archivos", null);
00950             list.append("Sincronizar", null);
00951             list.append("Ver archivos", null);
00952             list.append("Salir", null);
00953             list.setCommandListener(this);
00954             list.setSelectedFlags(new boolean[] { false, false, false, false, fal
se }); //GEN-END:|64-getter|1|64-postInit
00955             // write post-init user code here
00956             //probando la lectura de directorios

```

```

00957         //obtenerListadoArchivos("root1/");
00958         //creacionBitacoraLocal();
00959         }//GEN-BEGIN:|64-getter|2|
00960         return list;
00961     }
00962     //</editor-fold>//GEN-END:|64-getter|2|
00963
00964     //<editor-fold defaultstate="collapsed" desc=" Generated Method: listAction "
>//GEN-BEGIN:|64-action|0|64-preAction
00968     public void listAction() { //GEN-END:|64-action|0|64-preAction
00969         // enter pre-action user code here
00970         String __selectedString = getList().getString(getList().getSelectedIndex(
)); //GEN-BEGIN:|64-action|1|282-preAction
00971         if (__selectedString != null) {
00972             if (__selectedString.equals("Descargar archivos")) { //GEN-END:|64-act
ion|1|282-preAction
00973                 // write pre-action user code here
00974                 switchDisplayable(null, getForm()); //GEN-LINE:|64-action|2|282-po
stAction
00975                 // write post-action user code here
00976             } else if (__selectedString.equals("Subir archivos")) { //GEN-LINE:|64
-action|3|283-preAction
00977                 // write pre-action user code here
00978                 switchDisplayable(null, getForm1()); //GEN-LINE:|64-action|4|283-p
ostAction
00979                 // write post-action user code here
00980             } else if (__selectedString.equals("Sincronizar")) { //GEN-LINE:|64-ac
tion|5|379-preAction
00981                 // write pre-action user code here
00982                 switchDisplayable(null, getForm5()); //GEN-LINE:|64-action|6|379-p
ostAction
00983                 // write post-action user code here
00984             } else if (__selectedString.equals("Ver archivos")) { //GEN-LINE:|64-a
ction|7|71-preAction
00985                 // write pre-action user code here
00986                 switchDisplayable(null, getList2()); //GEN-LINE:|64-action|8|71-po
stAction
00987                 // write post-action user code here
00988             } else if (__selectedString.equals("Salir")) { //GEN-LINE:|64-action|9
|73-preAction
00989                 // write pre-action user code here
00990                 switchDisplayable(null, getSplashScreen()); //GEN-LINE:|64-action|
10|73-postAction
00991                 // write post-action user code here
00992             } //GEN-BEGIN:|64-action|11|64-postAction
00993         } //GEN-END:|64-action|11|64-postAction
00994         // enter post-action user code here
00995     } //GEN-BEGIN:|64-action|12|
00996     //</editor-fold>//GEN-END:|64-action|12|
00997
00998
00999     //<editor-fold defaultstate="collapsed" desc=" Generated Getter: okCommand2 "
>//GEN-BEGIN:|75-getter|0|75-preInit
01004     public Command getOkCommand2() {
01005         if (okCommand2 == null) { //GEN-END:|75-getter|0|75-preInit
01006             // write pre-init user code here
01007             okCommand2 = new Command("Ok", Command.OK, 0); //GEN-LINE:|75-getter|1

```

```

|75-postInit
01008     // write post-init user code here
01009     }//GEN-BEGIN:|75-getter|2|
01010     return okCommand2;
01011     }
01012     //</editor-fold>//GEN-END:|75-getter|2|
01013
01014
01015
01016     //<editor-fold defaultstate="collapsed" desc=" Generated Getter: okCommand3 "
>//GEN-BEGIN:|80-getter|0|80-preInit
01021     public Command getOkCommand3() {
01022         if (okCommand3 == null) { //GEN-END:|80-getter|0|80-preInit
01023             // write pre-init user code here
01024             okCommand3 = new Command("Ok", Command.OK, 0); //GEN-LINE:|80-getter|1
|80-postInit
01025         // write post-init user code here
01026         }//GEN-BEGIN:|80-getter|2|
01027         return okCommand3;
01028     }
01029     //</editor-fold>//GEN-END:|80-getter|2|
01030
01031     //<editor-fold defaultstate="collapsed" desc=" Generated Getter: okCommand4 "
>//GEN-BEGIN:|85-getter|0|85-preInit
01036     public Command getOkCommand4() {
01037         if (okCommand4 == null) { //GEN-END:|85-getter|0|85-preInit
01038             // write pre-init user code here
01039             okCommand4 = new Command("Ok", Command.OK, 0); //GEN-LINE:|85-getter|1
|85-postInit
01040         // write post-init user code here
01041         }//GEN-BEGIN:|85-getter|2|
01042         return okCommand4;
01043     }
01044     //</editor-fold>//GEN-END:|85-getter|2|
01045
01046
01047
01048
01049
01050
01051
01052     //<editor-fold defaultstate="collapsed" desc=" Generated Getter: okCommand5 "
>//GEN-BEGIN:|91-getter|0|91-preInit
01057     public Command getOkCommand5() {
01058         if (okCommand5 == null) { //GEN-END:|91-getter|0|91-preInit
01059             // write pre-init user code here
01060             okCommand5 = new Command("Ok", Command.OK, 0); //GEN-LINE:|91-getter|1
|91-postInit
01061         // write post-init user code here
01062         }//GEN-BEGIN:|91-getter|2|
01063         return okCommand5;
01064     }
01065     //</editor-fold>//GEN-END:|91-getter|2|
01066
01067     //<editor-fold defaultstate="collapsed" desc=" Generated Getter: fileBrowser
">//GEN-BEGIN:|89-getter|0|89-preInit
01072     public FileBrowser getFileBrowser() {

```



```

01073         if (fileBrowser == null) { //GEN-END:|89-getter|0|89-preInit
01074             // write pre-init user code here
01075             fileBrowser = new FileBrowser(getDisplay()); //GEN-BEGIN:|89-getter|1|
89-postInit
01076             fileBrowser.setTitle("Navegador de Archivos del m\u00F3vil");
01077             fileBrowser.setCommandListener(this);
01078             fileBrowser.addCommand(FileBrowser.SELECT_FILE_COMMAND);
01079             fileBrowser.addCommand(getBackCommand1()); //GEN-END:|89-getter|1|89-p
ostInit
01080             // write post-init user code here
01081             } //GEN-BEGIN:|89-getter|2|
01082             return fileBrowser;
01083         }
01084     } //</editor-fold> //GEN-END:|89-getter|2|
01085
01086     //<editor-fold defaultstate="collapsed" desc=" Generated Getter: backCommand
"> //GEN-BEGIN:|95-getter|0|95-preInit
01091     public Command getBackCommand() {
01092         if (backCommand == null) { //GEN-END:|95-getter|0|95-preInit
01093             // write pre-init user code here
01094             backCommand = new Command("Back", Command.BACK, 0); //GEN-LINE:|95-get
ter|1|95-postInit
01095             // write post-init user code here
01096             } //GEN-BEGIN:|95-getter|2|
01097             return backCommand;
01098         }
01099     } //</editor-fold> //GEN-END:|95-getter|2|
01100
01101     //<editor-fold defaultstate="collapsed" desc=" Generated Getter: okCommand6 "
"> //GEN-BEGIN:|104-getter|0|104-preInit
01106     public Command getOkCommand6() {
01107         if (okCommand6 == null) { //GEN-END:|104-getter|0|104-preInit
01108             // write pre-init user code here
01109             okCommand6 = new Command("OK", Command.OK, 0); //GEN-LINE:|104-getter|
1|104-postInit
01110             // write post-init user code here
01111             } //GEN-BEGIN:|104-getter|2|
01112             return okCommand6;
01113         }
01114     } //</editor-fold> //GEN-END:|104-getter|2|
01115
01116
01117
01118
01119
01120     //<editor-fold defaultstate="collapsed" desc=" Generated Getter: backCommand1
"> //GEN-BEGIN:|112-getter|0|112-preInit
01125     public Command getBackCommand1() {
01126         if (backCommand1 == null) { //GEN-END:|112-getter|0|112-preInit
01127             // write pre-init user code here
01128             backCommand1 = new Command("Back", Command.BACK, 0); //GEN-LINE:|112-g
etter|1|112-postInit
01129             // write post-init user code here
01130             } //GEN-BEGIN:|112-getter|2|
01131             return backCommand1;
01132         }
01133     } //</editor-fold> //GEN-END:|112-getter|2|

```



```

01134
01135     //<editor-fold defaultstate="collapsed" desc=" Generated Getter: exitCommand1
">//GEN-BEGIN:|115-getter|0|115-preInit
01140     public Command getExitCommand1() {
01141         if (exitCommand1 == null) { //GEN-END:|115-getter|0|115-preInit
01142             // write pre-init user code here
01143             exitCommand1 = new Command("Exit", Command.EXIT, 0); //GEN-LINE:|115-g
etter|1|115-postInit
01144             // write post-init user code here
01145             } //GEN-BEGIN:|115-getter|2|
01146             return exitCommand1;
01147         }
01148     } //</editor-fold> //GEN-END:|115-getter|2|
01149
01150     //<editor-fold defaultstate="collapsed" desc=" Generated Getter: list2 "> //GE
N-BEGIN:|117-getter|0|117-preInit
01155     public List getList2() {
01156         if (list2 == null) { //GEN-END:|117-getter|0|117-preInit
01157             // write pre-init user code here
01158             list2 = new List("Navegador de Archivos", Choice.IMPLICIT); //GEN-BEGI
N:|117-getter|1|117-postInit
01159             list2.append("Ver archivos del m\u00F3vil", null);
01160             list2.append("Ver archivos del servidor", null);
01161             list2.addCommand(getBackCommand3());
01162             list2.setCommandListener(this);
01163             list2.setSelectedFlags(new boolean[] { false, false }); //GEN-END:|117
-getter|1|117-postInit
01164             // write post-init user code here
01165             } //GEN-BEGIN:|117-getter|2|
01166             return list2;
01167         }
01168     } //</editor-fold> //GEN-END:|117-getter|2|
01169
01170     //<editor-fold defaultstate="collapsed" desc=" Generated Method: list2Action
"> //GEN-BEGIN:|117-action|0|117-preAction
01174     public void list2Action() { //GEN-END:|117-action|0|117-preAction
01175         // enter pre-action user code here
01176         String __selectedString = getList2().getString(getList2().getSelectedInde
x()); //GEN-BEGIN:|117-action|1|120-preAction
01177         if (__selectedString != null) {
01178             if (__selectedString.equals("Ver archivos del m\u00F3vil")) { //GEN-EN
D:|117-action|1|120-preAction
01179                 // write pre-action user code here
01180                 switchDisplayable(getVerArchivosMovil(), getFileBrowser()); //GEN-
LINE:|117-action|2|120-postAction
01181                 // write post-action user code here
01182             } else if (__selectedString.equals("Ver archivos del servidor")) { //G
EN-LINE:|117-action|3|121-preAction
01183                 // write pre-action user code here
01184                 switchDisplayable(getVerArchivosServidor(), getList3()); //GEN-LIN
E:|117-action|4|121-postAction
01185                 // write post-action user code here
01186             } //GEN-BEGIN:|117-action|5|117-postAction
01187             } //GEN-END:|117-action|5|117-postAction
01188             // enter post-action user code here
01189             } //GEN-BEGIN:|117-action|6|
01190             } //</editor-fold> //GEN-END:|117-action|6|

```

```

01191
01192     //<editor-fold defaultstate="collapsed" desc=" Generated Getter: list3 ">//GE
N-BEGIN:|124-getter|0|124-preInit
01197     public List getList3() {
01198         if (list3 == null) { //GEN-END:|124-getter|0|124-preInit
01199             // write pre-init user code here
01200             String rutaServicioWeb = "";
01201             String contenido = "";
01202
01203             Vector archivosRemoto = new Vector();
01204             StringTokenizer Token;
01205             int tok = 0;
01206             int rem = 0;
01207
01208             list3 = new List("Navegador de Archivos del Servidor", Choice.IMPLICI
T); //GEN-BEGIN:|124-getter|1|124-postInit
01209             list3.addCommand(getBackCommand2());
01210             list3.setCommandListener(this); //GEN-END:|124-getter|1|124-postInit
01211
01212             try{
01213
01214                 //creamos la ruta con la información que se va a enviar
01215                 rutaServicioWeb = urlDefinida + "/XMLTest/resources/todosarchivos
port/lista/?usuario=" +
01216                 usuarioActual + "&password=" + passwordActual;
01217
01218                 //recibimos la información del WS una vez que comparó la bitacora
01219                 local enviada
01220                 //con la bitocora del servidor
01221                 contenido = Parser.ConectorWeb.sendHttpGet(rutaServicioWeb);
01222
01223                 System.out.println("Lo que recibió de archivo:" + contenido + ":");
01224             };
01225
01226             Token = new StringTokenizer(contenido, "\n");
01227
01228             while (Token.hasMoreElements()){
01229                 //System.out.println(Token.nextElement());
01230                 archivosRemoto.addElement(Token.nextElement());
01231             }
01232             System.out.println(tok);
01233             tok = archivosRemoto.size();
01234
01235             boolean[] f = new boolean[tok];
01236
01237             for(int in = 0; in < tok; in++){
01238                 list3.append(archivosRemoto.firstElement().toString(), null);
01239
01240                 f[in]=false;
01241                 archivosRemoto.removeElementAt(rem);
01242             }
01243             list3.setSelectedFlags( f );
01244             }catch(Exception h){System.out.println(h);}
01245             // write post-init user code here
01246         } //GEN-BEGIN:|124-getter|2|
01247         return list3;

```

```

01246     }
01247     //</editor-fold>//GEN-END:|124-getter|2|
01248
01249     //<editor-fold defaultstate="collapsed" desc=" Generated Method: list3Action
">//GEN-BEGIN:|124-action|0|124-preAction
01253     public void list3Action() { //GEN-END:|124-action|0|124-preAction
01254         // enter pre-action user code here
01255         String __selectedString = getList3().getString(getList3().getSelectedInde
x()); //GEN-LINE:|124-action|1|124-postAction
01256         // enter post-action user code here
01257     } //GEN-BEGIN:|124-action|2|
01258     //</editor-fold>//GEN-END:|124-action|2|
01259
01260     //<editor-fold defaultstate="collapsed" desc=" Generated Getter: backCommand2
">//GEN-BEGIN:|128-getter|0|128-preInit
01265     public Command getBackCommand2() {
01266         if (backCommand2 == null) { //GEN-END:|128-getter|0|128-preInit
01267             // write pre-init user code here
01268             backCommand2 = new Command("Back", Command.BACK, 0); //GEN-LINE:|128-g
etter|1|128-postInit
01269             // write post-init user code here
01270         } //GEN-BEGIN:|128-getter|2|
01271         return backCommand2;
01272     }
01273     //</editor-fold>//GEN-END:|128-getter|2|
01274
01275     //<editor-fold defaultstate="collapsed" desc=" Generated Getter: backCommand3
">//GEN-BEGIN:|132-getter|0|132-preInit
01280     public Command getBackCommand3() {
01281         if (backCommand3 == null) { //GEN-END:|132-getter|0|132-preInit
01282             // write pre-init user code here
01283             backCommand3 = new Command("Back", Command.BACK, 0); //GEN-LINE:|132-g
etter|1|132-postInit
01284             // write post-init user code here
01285         } //GEN-BEGIN:|132-getter|2|
01286         return backCommand3;
01287     }
01288     //</editor-fold>//GEN-END:|132-getter|2|
01289
01290
01291
01292
01293
01294     //<editor-fold defaultstate="collapsed" desc=" Generated Getter: backCommand4
">//GEN-BEGIN:|148-getter|0|148-preInit
01299     public Command getBackCommand4() {
01300         if (backCommand4 == null) { //GEN-END:|148-getter|0|148-preInit
01301             // write pre-init user code here
01302             backCommand4 = new Command("Back", Command.BACK, 0); //GEN-LINE:|148-g
etter|1|148-postInit
01303             // write post-init user code here
01304         } //GEN-BEGIN:|148-getter|2|
01305         return backCommand4;
01306     }
01307     //</editor-fold>//GEN-END:|148-getter|2|
01308
01309     //<editor-fold defaultstate="collapsed" desc=" Generated Getter: waitScreen "

```



```

>//GEN-BEGIN:|152-getter|0|152-preInit
01314     public WaitScreen getWaitScreen() {
01315         if (waitScreen == null) { //GEN-END:|152-getter|0|152-preInit
01316             // write pre-init user code here
01317             waitScreen = new WaitScreen(getDisplay()); //GEN-BEGIN:|152-getter|1|1
52-postInit
01318             waitScreen.setTitle("waitScreen");
01319             waitScreen.setCommandListener(this);
01320             waitScreen.setTask(getTask1()); //GEN-END:|152-getter|1|152-postInit
01321             // write post-init user code here
01322         } //GEN-BEGIN:|152-getter|2|
01323         return waitScreen;
01324     }
01325     //</editor-fold> //GEN-END:|152-getter|2|
01326
01327
01328
01329     //<editor-fold defaultstate="collapsed" desc=" Generated Getter: task1 "> //GE
N-BEGIN:|155-getter|0|155-preInit
01334     public SimpleCancellableTask getTask1() {
01335         if (task1 == null) { //GEN-END:|155-getter|0|155-preInit
01336             // write pre-init user code here
01337             task1 = new SimpleCancellableTask(); //GEN-BEGIN:|155-getter|1|155-exe
cute
01338             task1.setExecutable(new org.netbeans.microedition.util.Executable() {
01339                 public void execute() throws Exception { //GEN-END:|155-getter|1|1
55-execute
01340                     // write task-execution user code here
01341                     creacionBitacora();
01342                     comparaBitacorasUp("bitacora.xml");
01343                     } //GEN-BEGIN:|155-getter|2|155-postInit
01344                 }); //GEN-END:|155-getter|2|155-postInit
01345                 // write post-init user code here
01346             } //GEN-BEGIN:|155-getter|3|
01347             return task1;
01348         }
01349     } //</editor-fold> //GEN-END:|155-getter|3|
01350
01351     //<editor-fold defaultstate="collapsed" desc=" Generated Getter: okCommand7 "
>//GEN-BEGIN:|159-getter|0|159-preInit
01356     public Command getOkCommand7() {
01357         if (okCommand7 == null) { //GEN-END:|159-getter|0|159-preInit
01358             // write pre-init user code here
01359             okCommand7 = new Command("OK", Command.OK, 0); //GEN-LINE:|159-getter|
1|159-postInit
01360             // write post-init user code here
01361         } //GEN-BEGIN:|159-getter|2|
01362         return okCommand7;
01363     }
01364     //</editor-fold> //GEN-END:|159-getter|2|
01365
01366     //<editor-fold defaultstate="collapsed" desc=" Generated Getter: bitacoraCrea
daUp "> //GEN-BEGIN:|162-getter|0|162-preInit
01371     public Alert getBitacoraCreadaUp() {
01372         if (bitacoraCreadaUp == null) { //GEN-END:|162-getter|0|162-preInit
01373             // write pre-init user code here

```

```

01374         bitacoraCreadaUp = new Alert("bitacoraok", "La conexi\u00F3n con el s
ervidor ha sido realizada", getImage2(), null);//GEN-BEGIN:|162-getter|1|162-post
Init
01375         bitacoraCreadaUp.setTimeout(2000);//GEN-END:|162-getter|1|162-postIni
t
01376         //bitacoraCreadaUp = new Alert("bitacoraok", "La conexi\u00F3n con el serv
idor ha sido realizada", null, null);
01377         // write post-init user code here
01378         }//GEN-BEGIN:|162-getter|2|
01379         return bitacoraCreadaUp;
01380     }
01381 }//</editor-fold>//GEN-END:|162-getter|2|
01382
01383 //<editor-fold defaultstate="collapsed" desc=" Generated Getter: bitacoraErro
rUp ">//GEN-BEGIN:|168-getter|0|168-preInit
01388 public Alert getBitacoraErrorUp() {
01389     if (bitacoraErrorUp == null) { //GEN-END:|168-getter|0|168-preInit
01390         // write pre-init user code here
01391         bitacoraErrorUp = new Alert("bitacoraerror", "La bitacora no pudo ser
creada", getImage3(), null);//GEN-BEGIN:|168-getter|1|168-postInit
01392         bitacoraErrorUp.setTimeout(2000);//GEN-END:|168-getter|1|168-postInit
01393
01394         //bitacoraErrorUp = new Alert("bitacoraerror", "La bitacora no pudo se
r creada", null, null);
01394         // write post-init user code here
01395         }//GEN-BEGIN:|168-getter|2|
01396         return bitacoraErrorUp;
01397     }
01398 }//</editor-fold>//GEN-END:|168-getter|2|
01399
01400
01401
01402
01403
01404 //<editor-fold defaultstate="collapsed" desc=" Generated Getter: okCommand8 "
>//GEN-BEGIN:|184-getter|0|184-preInit
01409 public Command getOkCommand8() {
01410     if (okCommand8 == null) { //GEN-END:|184-getter|0|184-preInit
01411         // write pre-init user code here
01412         okCommand8 = new Command("Ok", Command.OK, 0);//GEN-LINE:|184-getter|
1|184-postInit
01413         // write post-init user code here
01414         }//GEN-BEGIN:|184-getter|2|
01415         return okCommand8;
01416     }
01417 }//</editor-fold>//GEN-END:|184-getter|2|
01418
01419 //<editor-fold defaultstate="collapsed" desc=" Generated Getter: backCommand5
">//GEN-BEGIN:|188-getter|0|188-preInit
01424 public Command getBackCommand5() {
01425     if (backCommand5 == null) { //GEN-END:|188-getter|0|188-preInit
01426         // write pre-init user code here
01427         backCommand5 = new Command("Back", Command.BACK, 0);//GEN-LINE:|188-g
etter|1|188-postInit
01428         // write post-init user code here
01429         }//GEN-BEGIN:|188-getter|2|
01430         return backCommand5;

```

```

01431     }
01432     //</editor-fold>//GEN-END:|188-getter|2|
01433
01434     //<editor-fold defaultstate="collapsed" desc=" Generated Getter: backCommand6
">//GEN-BEGIN:|192-getter|0|192-preInit
01439     public Command getBackCommand6() {
01440         if (backCommand6 == null) { //GEN-END:|192-getter|0|192-preInit
01441             // write pre-init user code here
01442             backCommand6 = new Command("Back", Command.BACK, 0); //GEN-LINE:|192-g
etter|1|192-postInit
01443             // write post-init user code here
01444         } //GEN-BEGIN:|192-getter|2|
01445         return backCommand6;
01446     }
01447     //</editor-fold>//GEN-END:|192-getter|2|
01448
01449     //<editor-fold defaultstate="collapsed" desc=" Generated Getter: okCommand9 "
">//GEN-BEGIN:|195-getter|0|195-preInit
01454     public Command getOkCommand9() {
01455         if (okCommand9 == null) { //GEN-END:|195-getter|0|195-preInit
01456             // write pre-init user code here
01457             okCommand9 = new Command("OK", Command.OK, 0); //GEN-LINE:|195-getter|
1|195-postInit
01458             // write post-init user code here
01459         } //GEN-BEGIN:|195-getter|2|
01460         return okCommand9;
01461     }
01462     //</editor-fold>//GEN-END:|195-getter|2|
01463
01464     //<editor-fold defaultstate="collapsed" desc=" Generated Getter: okCommand10
">//GEN-BEGIN:|197-getter|0|197-preInit
01469     public Command getOkCommand10() {
01470         if (okCommand10 == null) { //GEN-END:|197-getter|0|197-preInit
01471             // write pre-init user code here
01472             okCommand10 = new Command("Ok", Command.OK, 0); //GEN-LINE:|197-getter
1|197-postInit
01473             // write post-init user code here
01474         } //GEN-BEGIN:|197-getter|2|
01475         return okCommand10;
01476     }
01477     //</editor-fold>//GEN-END:|197-getter|2|
01478
01479     //<editor-fold defaultstate="collapsed" desc=" Generated Getter: okCommand11
">//GEN-BEGIN:|204-getter|0|204-preInit
01484     public Command getOkCommand11() {
01485         if (okCommand11 == null) { //GEN-END:|204-getter|0|204-preInit
01486             // write pre-init user code here
01487             okCommand11 = new Command("Ok", Command.OK, 0); //GEN-LINE:|204-getter
1|204-postInit
01488             // write post-init user code here
01489         } //GEN-BEGIN:|204-getter|2|
01490         return okCommand11;
01491     }
01492     //</editor-fold>//GEN-END:|204-getter|2|
01493
01494
01495

```



```

01496     //<editor-fold defaultstate="collapsed" desc=" Generated Getter: okCommand12
">//GEN-BEGIN:|215-getter|0|215-preInit
01501     public Command getOkCommand12() {
01502         if (okCommand12 == null) { //GEN-END:|215-getter|0|215-preInit
01503             // write pre-init user code here
01504             okCommand12 = new Command("Ok", Command.OK, 0); //GEN-LINE:|215-getter
|1|215-postInit
01505             // write post-init user code here
01506         } //GEN-BEGIN:|215-getter|2|
01507         return okCommand12;
01508     }
01509     //</editor-fold>//GEN-END:|215-getter|2|
01510
01511     //<editor-fold defaultstate="collapsed" desc=" Generated Getter: textBox1 ">/
/GEN-BEGIN:|218-getter|0|218-preInit
01516     public TextBox getTextBox1() {
01517         if (textBox1 == null) { //GEN-END:|218-getter|0|218-preInit
01518             // write pre-init user code here
01519             textBox1 = new TextBox("Inicio de Transferencia de Archivos", "\n\n\n
\n\n\n\n Presione OK para continuar...", 100, TextField.ANY | TextField.UNED
ITABLE); //GEN-BEGIN:|218-getter|1|218-postInit
01520             textBox1.addCommand(getOkCommand13());
01521             textBox1.addCommand(getCancelCommand1());
01522             textBox1.setCommandListener(this); //GEN-END:|218-getter|1|218-postIni
t
01523             // write post-init user code here
01524         } //GEN-BEGIN:|218-getter|2|
01525         return textBox1;
01526     }
01527     //</editor-fold>//GEN-END:|218-getter|2|
01528
01529     //<editor-fold defaultstate="collapsed" desc=" Generated Getter: okCommand13
">//GEN-BEGIN:|219-getter|0|219-preInit
01534     public Command getOkCommand13() {
01535         if (okCommand13 == null) { //GEN-END:|219-getter|0|219-preInit
01536             // write pre-init user code here
01537             okCommand13 = new Command("Ok", Command.OK, 0); //GEN-LINE:|219-getter
|1|219-postInit
01538             // write post-init user code here
01539         } //GEN-BEGIN:|219-getter|2|
01540         return okCommand13;
01541     }
01542     //</editor-fold>//GEN-END:|219-getter|2|
01543
01544     //<editor-fold defaultstate="collapsed" desc=" Generated Getter: waitScreen1
">//GEN-BEGIN:|223-getter|0|223-preInit
01549     public WaitScreen getWaitScreen1() {
01550         if (waitScreen1 == null) { //GEN-END:|223-getter|0|223-preInit
01551             // write pre-init user code here
01552             waitScreen1 = new WaitScreen(getDisplay()); //GEN-BEGIN:|223-getter|1|
223-postInit
01553             waitScreen1.setTitle("waitScreen1");
01554             waitScreen1.setCommandListener(this);
01555             waitScreen1.setTask(getTask2()); //GEN-END:|223-getter|1|223-postInit
01556             // write post-init user code here
01557         } //GEN-BEGIN:|223-getter|2|
01558         return waitScreen1;

```



```

01559     }
01560     //</editor-fold>//GEN-END:|223-getter|2|
01561
01562     //<editor-fold defaultstate="collapsed" desc=" Generated Getter: transferenci
aCompletaUp ">//GEN-BEGIN:|228-getter|0|228-preInit
01567     public Alert getTransferenciaCompletaUp() {
01568         if (transferenciaCompletaUp == null) { //GEN-END:|228-getter|0|228-preInit

01569             // write pre-init user code here
01570             transferenciaCompletaUp = new Alert("transferenciaok", "La transferen
cia se ha completado", getImage2(), null); //GEN-BEGIN:|228-getter|1|228-postInit
01571             transferenciaCompletaUp.setTimeout(2000); //GEN-END:|228-getter|1|228-
postInit
01572             //transferenciaCompletaUp = new Alert("transferenciaok", "La transfer
encia se ha completado", null, null);
01573             // write post-init user code here
01574             } //GEN-BEGIN:|228-getter|2|
01575             return transferenciaCompletaUp;
01576         }
01577     //</editor-fold>//GEN-END:|228-getter|2|
01578
01579     //<editor-fold defaultstate="collapsed" desc=" Generated Getter: task2 ">//GE
N-BEGIN:|226-getter|0|226-preInit
01584     public SimpleCancellableTask getTask2() {
01585         if (task2 == null) { //GEN-END:|226-getter|0|226-preInit
01586             // write pre-init user code here
01587             task2 = new SimpleCancellableTask(); //GEN-BEGIN:|226-getter|1|226-exe
cute
01588             task2.setExecutable(new org.netbeans.microedition.util.Executable() {

01589                 public void execute() throws Exception { //GEN-END:|226-getter|1|2
26-execute
01590                     Vector salidaTransferencia2;
01591
01592                     salidaTransferencia2 = realizarTransferencial();
01593                     salidaSeleccion2 = seleccionDeArchivos(salidaTransferencia2);

01594                     // write task-execution user code here
01595                     } //GEN-BEGIN:|226-getter|2|226-postInit
01596                 }); //GEN-END:|226-getter|2|226-postInit
01597                 // write post-init user code here
01598                 } //GEN-BEGIN:|226-getter|3|
01599                 return task2;
01600             }
01601     //</editor-fold>//GEN-END:|226-getter|3|
01602
01603     //<editor-fold defaultstate="collapsed" desc=" Generated Getter: transferenci
aErrorUp ">//GEN-BEGIN:|230-getter|0|230-preInit
01608     public Alert getTransferenciaErrorUp() {
01609         if (transferenciaErrorUp == null) { //GEN-END:|230-getter|0|230-preInit
01610             // write pre-init user code here
01611             transferenciaErrorUp = new Alert("transferenciaincompleta", "La trans
ferencia no se realiz\u00F3 con \u00E9xito", getImage3(), null); //GEN-BEGIN:|230-
getter|1|230-postInit
01612             transferenciaErrorUp.setTimeout(2000); //GEN-END:|230-getter|1|230-pos
tInit
01613             //transferenciaErrorUp = new Alert("transferenciafail", "La transfere

```

```

        ncia no se realizo con exito", null, null);
01614         // write post-init user code here
01615     }//GEN-BEGIN:|230-getter|2|
01616     return transferenciaErrorUp;
01617 }
01618 //</editor-fold>//GEN-END:|230-getter|2|
01619
01620 //<editor-fold defaultstate="collapsed" desc=" Generated Getter: okCommand14
">//GEN-BEGIN:|233-getter|0|233-preInit
01625     public Command getOkCommand14() {
01626         if (okCommand14 == null) { //GEN-END:|233-getter|0|233-preInit
01627             // write pre-init user code here
01628             okCommand14 = new Command("Ok", Command.OK, 0); //GEN-LINE:|233-getter
|1|233-postInit
01629             // write post-init user code here
01630         } //GEN-BEGIN:|233-getter|2|
01631         return okCommand14;
01632     }
01633 //</editor-fold>//GEN-END:|233-getter|2|
01634
01635
01636
01637 //<editor-fold defaultstate="collapsed" desc=" Generated Getter: waitScreen2
">//GEN-BEGIN:|236-getter|0|236-preInit
01642     public WaitScreen getWaitScreen2() {
01643         if (waitScreen2 == null) { //GEN-END:|236-getter|0|236-preInit
01644             // write pre-init user code here
01645             waitScreen2 = new WaitScreen(getDisplay()); //GEN-BEGIN:|236-getter|1|
236-postInit
01646             waitScreen2.setTitle("waitScreen2");
01647             waitScreen2.setCommandListener(this);
01648             waitScreen2.setTask(getTask3()); //GEN-END:|236-getter|1|236-postInit
01649             // write post-init user code here
01650         } //GEN-BEGIN:|236-getter|2|
01651         return waitScreen2;
01652     }
01653 //</editor-fold>//GEN-END:|236-getter|2|
01654
01655 //<editor-fold defaultstate="collapsed" desc=" Generated Getter: bitacoraCrea
daDown ">//GEN-BEGIN:|241-getter|0|241-preInit
01660     public Alert getBitacoraCreadaDown() {
01661         if (bitacoraCreadaDown == null) { //GEN-END:|241-getter|0|241-preInit
01662             // write pre-init user code here
01663             bitacoraCreadaDown = new Alert("bitacoraok", "La conexi\u00F3n con el
servidor ha sido realizada", getImage2(), null); //GEN-BEGIN:|241-getter|1|241-po
stInit
01664             bitacoraCreadaDown.setTimeout(2000); //GEN-END:|241-getter|1|241-postI
nit
01665             //bitacoraCreadaDown = new Alert("bitacoraok", "La conexi\u00F3n con el se
rvidor ha sido realizada", null, null);
01666             // write post-init user code here
01667         } //GEN-BEGIN:|241-getter|2|
01668         return bitacoraCreadaDown;
01669     }
01670 //</editor-fold>//GEN-END:|241-getter|2|
01671
01672 //<editor-fold defaultstate="collapsed" desc=" Generated Getter: bitacoraErro

```



```

01731
01732
01733
01734
01739     public Command getOkCommand25() {
01740         if (okCommand25 == null) {
01741             // write pre-init user code here
01742             okCommand25 = new Command("Ok", Command.OK, 0);
01743             // write post-init user code here
01744         }
01745         return okCommand25;
01746     }
01747
01752     public Command getCancelCommand25() {
01753         if (cancelCommand25 == null) {
01754             // write pre-init user code here
01755             cancelCommand25 = new Command("Cancel", Command.CANCEL, 0);
01756             // write post-init user code here
01757         }
01758         return cancelCommand25;
01759     }
01760
01761
01766     public TextBox getAlertaArchivosCS() {
01767         if (textBoxAlertaCS == null) {
01768             // write pre-init user code here
01769             String texto = new String("\n\n\n\n\n\n\nEl archivo " + nombreAlert
a +
01770                 " ya existe en el servidor y el cliente.\n" +
01771                 "Qué versión desea resguardar ?");
01772             textBoxAlertaCS = new TextBox("Ya Existe el archivo", texto, 300, Tex
tField.ANY | TextField.UNEDITABLE);
01773             textBoxAlertaCS.addCommand(getOkCommand30());
01774             textBoxAlertaCS.addCommand(getCancelCommand30());
01775             textBoxAlertaCS.setCommandListener(this);
01776
01777             // write post-init user code here
01778         }
01779         else if (textBoxAlertaCS != null)
01780         {
01781             String texto = new String("\n\n\n\n\n\n\nEl archivo " + nombreA
lerta +
01782                 " ya existe en el servidor y el cliente.\n" +
01783                 "Qué versión desea resguardar ?");
01784             textBoxAlertaCS.setString(texto);
01785         }
01786
01787         return textBoxAlertaCS;
01788     }
01789
01790
01795     //Este OK será tomado como "Servidor"
01796     public Command getOkCommand30() {
01797         if (okCommand30 == null) {
01798             // write pre-init user code here
01799             okCommand30 = new Command("Servidor", Command.OK, 0);
01800             // write post-init user code here

```

```

01801     }
01802     return okCommand30;
01803 }
01804
01809 //Este Cancel será tomado como "Cliente"
01810 public Command getCancelCommand30() {
01811     if (cancelCommand30 == null) {
01812         // write pre-init user code here
01813         cancelCommand30 = new Command("Cliente", Command.CANCEL, 0);
01814         // write post-init user code here
01815     }
01816     return cancelCommand30;
01817 }
01818
01819 //<editor-fold defaultstate="collapsed" desc=" Generated Getter: task3 ">//GE
N-BEGIN:|239-getter|0|239-preInit
01824 public SimpleCancellableTask getTask3() {
01825     if (task3 == null) { //GEN-END:|239-getter|0|239-preInit
01826         // write pre-init user code here
01827         task3 = new SimpleCancellableTask(); //GEN-BEGIN:|239-getter|1|239-exe
cute
01828         task3.setExecutable(new org.netbeans.microedition.util.Executable() {
01829
01830             public void execute() throws Exception { //GEN-END:|239-getter|1|2
39-execute
01830                 creacionBitacora();
01831                 comparaBitacorasDown("bitacora.xml");
01832                 // write task-execution user code here
01833                 } //GEN-BEGIN:|239-getter|2|239-postInit
01834             }); //GEN-END:|239-getter|2|239-postInit
01835             // write post-init user code here
01836             } //GEN-BEGIN:|239-getter|3|
01837             return task3;
01838         }
01839     } //</editor-fold> //GEN-END:|239-getter|3|
01840
01841 //<editor-fold defaultstate="collapsed" desc=" Generated Getter: okCommand15
"> //GEN-BEGIN:|246-getter|0|246-preInit
01846 public Command getOkCommand15() {
01847     if (okCommand15 == null) { //GEN-END:|246-getter|0|246-preInit
01848         // write pre-init user code here
01849         okCommand15 = new Command("Ok", Command.OK, 0); //GEN-LINE:|246-getter
|1|246-postInit
01850         // write post-init user code here
01851     } //GEN-BEGIN:|246-getter|2|
01852     return okCommand15;
01853 }
01854 //</editor-fold> //GEN-END:|246-getter|2|
01855
01856 //<editor-fold defaultstate="collapsed" desc=" Generated Getter: waitScreen3
"> //GEN-BEGIN:|248-getter|0|248-preInit
01861 public WaitScreen getWaitScreen3() {
01862     if (waitScreen3 == null) { //GEN-END:|248-getter|0|248-preInit
01863         // write pre-init user code here
01864         waitScreen3 = new WaitScreen(getDisplay()); //GEN-BEGIN:|248-getter|1|
248-postInit
01865         waitScreen3.setTitle("waitScreen3");

```

```

01866         waitScreen3.setCommandListener(this);
01867         waitScreen3.setTask(getTask4()); //GEN-END:|248-getter|1|248-postInit
01868         // write post-init user code here
01869     } //GEN-BEGIN:|248-getter|2|
01870     return waitScreen3;
01871 }
01872 //</editor-fold> //GEN-END:|248-getter|2|
01873
01874 //<editor-fold defaultstate="collapsed" desc=" Generated Getter: transferencia
aCompletaDown "> //GEN-BEGIN:|253-getter|0|253-preInit
01879     public Alert getTransferenciaCompletaDown() {
01880         if (transferenciaCompletaDown == null) { //GEN-END:|253-getter|0|253-preIn
it
01881             // write pre-init user code here
01882
01883             transferenciaCompletaDown = new Alert("transferenciacompleta", "La tr
ansferencia se ha completado", getImage2(), null); //GEN-BEGIN:|253-getter|1|253-p
ostInit
01884             transferenciaCompletaDown.setTimeout(2000); //GEN-END:|253-getter|1|25
3-postInit
01885
01886
01887
01888             // write post-init user code here
01889         } //GEN-BEGIN:|253-getter|2|
01890         return transferenciaCompletaDown;
01891     }
01892 //</editor-fold> //GEN-END:|253-getter|2|
01893
01894 //<editor-fold defaultstate="collapsed" desc=" Generated Getter: transferencia
aErrorDown "> //GEN-BEGIN:|254-getter|0|254-preInit
01899     public Alert getTransferenciaErrorDown() {
01900         if (transferenciaErrorDown == null) { //GEN-END:|254-getter|0|254-preInit
// write pre-init user code here
01901             transferenciaErrorDown = new Alert("transferenciaincompleta", "La tra
nsferencia no se realiz\u00F3 con \u00E9xito", getImage3(), null); //GEN-BEGIN:|25
4-getter|1|254-postInit
01902             transferenciaErrorDown.setTimeout(2000); //GEN-END:|254-getter|1|254-p
ostInit
01903
01904             //transferenciaErrorDown = new Alert("transferenciaincompleta", "La t
ransferencia no se realizo con exito", null, null);
01905             // write post-init user code here
01906         } //GEN-BEGIN:|254-getter|2|
01907         return transferenciaErrorDown;
01908     }
01909 //</editor-fold> //GEN-END:|254-getter|2|
01910
01911 //<editor-fold defaultstate="collapsed" desc=" Generated Getter: task4 "> //GE
N-BEGIN:|251-getter|0|251-preInit
01916     public SimpleCancellableTask getTask4() {
01917         if (task4 == null) { //GEN-END:|251-getter|0|251-preInit
// write pre-init user code here
01918             task4 = new SimpleCancellableTask(); //GEN-BEGIN:|251-getter|1|251-exe
cute
01919
01920             task4.setExecutable(new org.netbeans.microedition.util.Executable() {
01921
01922                 public void execute() throws Exception { //GEN-END:|251-getter|1|2

```



```

51-execute
01922         Vector salidaTransferencia;
01923
01924         salidaTransferencia = realizarTransferencia();
01925         salidaSeleccion1 = seleccionDeArchivos(salidaTransferencia);
01926
01927         // write task-execution user code here
01928         }//GEN-BEGIN:|251-getter|2|251-postInit
01929     });//GEN-END:|251-getter|2|251-postInit
01930     // write post-init user code here
01931     }//GEN-BEGIN:|251-getter|3|
01932     return task4;
01933 }
01934 //</editor-fold>//GEN-END:|251-getter|3|
01935
01936 //<editor-fold defaultstate="collapsed" desc=" Generated Getter: form ">//GEN
-BEGIN:|259-getter|0|259-preInit
01941     public Form getForm() {
01942         if (form == null) { //GEN-END:|259-getter|0|259-preInit
01943             // write pre-init user code here
01944
01945             form = new Form("Opciones de sincronizaci\u00F3n", new Item[] {
01946                 getChoiceGroup() }); //GEN-BEGIN:|259-getter|1|259-postInit
01947             form.addCommand(getOkCommand16());
01948             form.addCommand(getBackCommand7());
01949             form.setCommandListener(this); //GEN-END:|259-getter|1|259-postInit
01950             // write post-init user code here
01951             } //GEN-BEGIN:|259-getter|2|
01952             return form;
01953         }
01954     } //</editor-fold>//GEN-END:|259-getter|2|
01955
01956 //<editor-fold defaultstate="collapsed" desc=" Generated Getter: choiceGroup
">//GEN-BEGIN:|260-getter|0|260-preInit
01960     public ChoiceGroup getChoiceGroup() {
01961         if (choiceGroup == null) { //GEN-END:|260-getter|0|260-preInit
01962             // write pre-init user code here
01963             choiceGroup = new ChoiceGroup("", Choice.EXCLUSIVE); //GEN-BEGIN:|260-
01964                 getter|1|260-postInit
01965             choiceGroup.append("Sincronizaci\u00F3n sin reemplazo", null);
01966             choiceGroup.append("Sincronizaci\u00F3n con reemplazo", null);
01967             choiceGroup.append("Sincronizaci\u00F3n con reemplazo y preguntar en
01968                 caso de reemplazo", null);
01969             choiceGroup.setFitPolicy(Choice.TEXT_WRAP_DEFAULT);
01970             choiceGroup.setSelectedFlags(new boolean[] { true, false, false }); //
01971                 GEN-END:|260-getter|1|260-postInit
01972             // write post-init user code here
01973             } //GEN-BEGIN:|260-getter|2|
01974             return choiceGroup;
01975         }
01976     } //</editor-fold>//GEN-END:|260-getter|2|
01977
01978 //<editor-fold defaultstate="collapsed" desc=" Generated Getter: okCommand16
">//GEN-BEGIN:|267-getter|0|267-preInit

```



```

01983     public Command getOkCommand16() {
01984         if (okCommand16 == null) { //GEN-END:|267-getter|0|267-preInit
01985             // write pre-init user code here
01986             okCommand16 = new Command("Ok", Command.OK, 0); //GEN-LINE:|267-getter
|1|267-postInit
01987             // write post-init user code here
01988             } //GEN-BEGIN:|267-getter|2|
01989             return okCommand16;
01990         }
01991     } //</editor-fold> //GEN-END:|267-getter|2|
01992
01993
01994
01995     //<editor-fold defaultstate="collapsed" desc=" Generated Getter: okCommand17
"> //GEN-BEGIN:|275-getter|0|275-preInit
02000     public Command getOkCommand17() {
02001         if (okCommand17 == null) { //GEN-END:|275-getter|0|275-preInit
02002             // write pre-init user code here
02003             okCommand17 = new Command("Ok", Command.OK, 0); //GEN-LINE:|275-getter
|1|275-postInit
02004             // write post-init user code here
02005             } //GEN-BEGIN:|275-getter|2|
02006             return okCommand17;
02007         }
02008     } //</editor-fold> //GEN-END:|275-getter|2|
02009
02010     //<editor-fold defaultstate="collapsed" desc=" Generated Getter: form1 "> //GE
N-BEGIN:|274-getter|0|274-preInit
02015     public Form getForm1() {
02016         if (form1 == null) { //GEN-END:|274-getter|0|274-preInit
02017             // write pre-init user code here
02018             form1 = new Form("Opciones de sincronizaci\u00F3n", new Item[] {
getChoiceGroup1() }); //GEN-BEGIN:|274-getter|1|274-postInit
02019             form1.addCommand(getOkCommand17());
02020             form1.addCommand(getBackCommand8());
02021             form1.setCommandListener(this); //GEN-END:|274-getter|1|274-postInit
02022             // write post-init user code here
02023             } //GEN-BEGIN:|274-getter|2|
02024             return form1;
02025         }
02026     } //</editor-fold> //GEN-END:|274-getter|2|
02027
02028     //<editor-fold defaultstate="collapsed" desc=" Generated Getter: choiceGroup1
"> //GEN-BEGIN:|279-getter|0|279-preInit
02033     public ChoiceGroup getChoiceGroup1() {
02034         if (choiceGroup1 == null) { //GEN-END:|279-getter|0|279-preInit
02035             // write pre-init user code here
02036             choiceGroup1 = new ChoiceGroup("", Choice.EXCLUSIVE); //GEN-BEGIN:|279
-getter|1|279-postInit
02037             choiceGroup1.append("Sincronizaci\u00F3n sin reemplazo", null);
02038             choiceGroup1.append("Sincronizaci\u00F3n con reemplazo", null);
02039             choiceGroup1.append("Sincronizaci\u00F3n con reemplazo y preguntar en
caso de reemplazo", null);
02040             choiceGroup1.setSelectedFlags(new boolean[] { true, false, false }); //
/GEN-END:|279-getter|1|279-postInit
02041             // write post-init user code here
02042             } //GEN-BEGIN:|279-getter|2|

```

```

02043         return choiceGroup1;
02044     }
02045     //</editor-fold>//GEN-END:|279-getter|2|
02046
02047     //<editor-fold defaultstate="collapsed" desc=" Generated Getter: backCommand7
">//GEN-BEGIN:|286-getter|0|286-preInit
02052     public Command getBackCommand7() {
02053         if (backCommand7 == null) { //GEN-END:|286-getter|0|286-preInit
02054             // write pre-init user code here
02055             backCommand7 = new Command("Back", Command.BACK, 0); //GEN-LINE:|286-g
etter|1|286-postInit
02056             // write post-init user code here
02057         } //GEN-BEGIN:|286-getter|2|
02058         return backCommand7;
02059     }
02060     //</editor-fold>//GEN-END:|286-getter|2|
02061
02062     //<editor-fold defaultstate="collapsed" desc=" Generated Getter: backCommand8
">//GEN-BEGIN:|289-getter|0|289-preInit
02067     public Command getBackCommand8() {
02068         if (backCommand8 == null) { //GEN-END:|289-getter|0|289-preInit
02069             // write pre-init user code here
02070             backCommand8 = new Command("Back", Command.BACK, 0); //GEN-LINE:|289-g
etter|1|289-postInit
02071             // write post-init user code here
02072         } //GEN-BEGIN:|289-getter|2|
02073         return backCommand8;
02074     }
02075     //</editor-fold>//GEN-END:|289-getter|2|
02076
02077
02078
02079
02080
02081     //<editor-fold defaultstate="collapsed" desc=" Generated Getter: itemCommand
">//GEN-BEGIN:|294-getter|0|294-preInit
02086     public Command getItemCommand() {
02087         if (itemCommand == null) { //GEN-END:|294-getter|0|294-preInit
02088             // write pre-init user code here
02089             itemCommand = new Command("Configurar Servidor", Command.ITEM, 0); //G
EN-LINE:|294-getter|1|294-postInit
02090             // write post-init user code here
02091         } //GEN-BEGIN:|294-getter|2|
02092         return itemCommand;
02093     }
02094     //</editor-fold>//GEN-END:|294-getter|2|
02095
02096
02097
02098     //<editor-fold defaultstate="collapsed" desc=" Generated Getter: itemCommand1
">//GEN-BEGIN:|303-getter|0|303-preInit
02103     public Command getItemCommand1() {
02104         if (itemCommand1 == null) { //GEN-END:|303-getter|0|303-preInit
02105             // write pre-init user code here
02106             itemCommand1 = new Command("Guardar", Command.ITEM, 0); //GEN-LINE:|30
3-getter|1|303-postInit
02107             // write post-init user code here

```

```

02108         } //GEN-BEGIN:|303-getter|2|
02109         return itemCommand1;
02110     }
02111     //</editor-fold> //GEN-END:|303-getter|2|
02112
02113     //<editor-fold defaultstate="collapsed" desc=" Generated Getter: backCommand9
"> //GEN-BEGIN:|305-getter|0|305-preInit
02118     public Command getBackCommand9() {
02119         if (backCommand9 == null) { //GEN-END:|305-getter|0|305-preInit
02120             // write pre-init user code here
02121             backCommand9 = new Command("Cancelar", Command.BACK, 0); //GEN-LINE:|3
05-getter|1|305-postInit
02122             // write post-init user code here
02123         } //GEN-BEGIN:|305-getter|2|
02124         return backCommand9;
02125     }
02126     //</editor-fold> //GEN-END:|305-getter|2|
02127
02128     //<editor-fold defaultstate="collapsed" desc=" Generated Getter: form2 "> //GE
N-BEGIN:|300-getter|0|300-preInit
02133     public Form getForm2() {
02134         if (form2 == null) { //GEN-END:|300-getter|0|300-preInit
02135             // write pre-init user code here
02136             form2 = new Form("Configuraci\u00F3n de URL", new Item[] {
getTextField() }); //GEN-BEGIN:|300-getter|1|300-postInit
02137             form2.addCommand(getItemCommand1());
02138             form2.addCommand(getBackCommand9());
02139             form2.setCommandListener(this); //GEN-END:|300-getter|1|300-postInit
02140             System.out.println("la cajita dice: " + textField.getString().toStrin
g());
02141             // write post-init user code here
02142         } //GEN-BEGIN:|300-getter|2|
02143         return form2;
02144     }
02145     //</editor-fold> //GEN-END:|300-getter|2|
02146
02147
02148
02149     //<editor-fold defaultstate="collapsed" desc=" Generated Getter: textField ">
//GEN-BEGIN:|302-getter|0|302-preInit
02154     public TextField getTextField() {
02155         if (textField == null) { //GEN-END:|302-getter|0|302-preInit
02156             // write pre-init user code here
02157
02158             String url = "propiedades.xml";
02159             String getUrl = "";
02160             textField = new TextField("Escribe la direcci\u00F3n del servidor de
Archivos", null, 32, TextField.ANY); //GEN-LINE:|302-getter|1|302-postInit
02161             Document docurltexto = Parser.RestParser.getDocumentLocal("file:///ro
ot/" + url);
02162             getUrl = Parser.RestParser.beginParsePropiedad(docurltexto).getUrlser
vidor().toString();
02163             textField.setString(getUrl);
02164             // write post-init user code here
02165         } //GEN-BEGIN:|302-getter|2|
02166         return textField;
02167     }

```



```

02168     //</editor-fold>//GEN-END:|302-getter|2|
02169
02170     //<editor-fold defaultstate="collapsed" desc=" Generated Getter: textBox6 ">/
//GEN-BEGIN:|310-getter|0|310-preInit
02175     public TextBox getTextBox6() {
02176         if (textBox6 == null) { //GEN-END:|310-getter|0|310-preInit
02177             // write pre-init user code here
02178             textBox6 = new TextBox("Guardar", "Tu informaci\u00F3n del servidor h
a sido guardada.", 100, TextField.ANY | TextField.UNEDITABLE); //GEN-BEGIN:|310-ge
tter|1|310-postInit
02179                 textBox6.addCommand(getOkCommand18());
02180                 textBox6.setCommandListener(this); //GEN-END:|310-getter|1|310-postIni
t
02181                 // write post-init user code here
02182             } //GEN-BEGIN:|310-getter|2|
02183             return textBox6;
02184         }
02185     //</editor-fold>//GEN-END:|310-getter|2|
02186
02187     //<editor-fold defaultstate="collapsed" desc=" Generated Getter: okCommand18
">//GEN-BEGIN:|312-getter|0|312-preInit
02192     public Command getOkCommand18() {
02193         if (okCommand18 == null) { //GEN-END:|312-getter|0|312-preInit
02194             // write pre-init user code here
02195             okCommand18 = new Command("Ok", Command.OK, 0); //GEN-LINE:|312-getter
|1|312-postInit
02196                 // write post-init user code here
02197             } //GEN-BEGIN:|312-getter|2|
02198             return okCommand18;
02199         }
02200     //</editor-fold>//GEN-END:|312-getter|2|
02201
02202     //<editor-fold defaultstate="collapsed" desc=" Generated Getter: form3 ">//GE
N-BEGIN:|315-getter|0|315-preInit
02207     public Form getForm3() {
02208         if (form3 == null) { //GEN-END:|315-getter|0|315-preInit
02209             // write pre-init user code here
02210             form3 = new Form("Descarga", new Item[] { getChoiceGroup2() }); //GEN-
BEGIN:|315-getter|1|315-postInit
02211                 form3.addCommand(getItemCommand2());
02212                 form3.addCommand(getItemCommand3());
02213                 form3.addCommand(getOkCommand19());
02214                 form3.setCommandListener(this); //GEN-END:|315-getter|1|315-postInit
02215                 // write post-init user code here
02216             } //GEN-BEGIN:|315-getter|2|
02217             return form3;
02218         }
02219     //</editor-fold>//GEN-END:|315-getter|2|
02220
02221     //<editor-fold defaultstate="collapsed" desc=" Generated Getter: okCommand19
">//GEN-BEGIN:|317-getter|0|317-preInit
02226     public Command getOkCommand19() {
02227         if (okCommand19 == null) { //GEN-END:|317-getter|0|317-preInit
02228             // write pre-init user code here
02229             okCommand19 = new Command("Ok", Command.OK, 0); //GEN-LINE:|317-getter
|1|317-postInit
02230                 // write post-init user code here

```

```

02231         } //GEN-BEGIN:|317-getter|2|
02232         return okCommand19;
02233     }
02234     //</editor-fold> //GEN-END:|317-getter|2|
02235
02236     //<editor-fold defaultstate="collapsed" desc=" Generated Getter: choiceGroup2
"> //GEN-BEGIN:|319-getter|0|319-preInit
02241     public ChoiceGroup getChoiceGroup2() {
02242         if (choiceGroup2 == null) { //GEN-END:|319-getter|0|319-preInit
02243             // write pre-init user code here
02244             String tmpAgrega = "";
02245             int q = 0;
02246             int size = 0;
02247             size = salidaSeleccion1.size();
02248             boolean[] f = new boolean[size];
02249
02250             choiceGroup2 = new ChoiceGroup("Selecciona los archivos a sincronizar
", Choice.MULTIPLE); //GEN-BEGIN:|319-getter|1|319-postInit
02251             choiceGroup2.setSelectedFlags(new boolean[] { }); //GEN-END:|319-gett
er|1|319-postInit
02252
02253             while(!salidaSeleccion1.isEmpty()){
02254
02255                 tmpAgrega = salidaSeleccion1.firstElement().toString();
02256                 choiceGroup2.append(tmpAgrega.toString(), null);
02257                 salidaSeleccion1.removeElementAt(q);
02258             }
02259
02260             for(int in = 0; in < size; in++){
02261                 f[in]=false;
02262             }
02263
02264             choiceGroup2.setSelectedFlags(f);
02265
02266             // write post-init user code here
02267         } //GEN-BEGIN:|319-getter|2|
02268         return choiceGroup2;
02269     }
02270     //</editor-fold> //GEN-END:|319-getter|2|
02271
02272     //<editor-fold defaultstate="collapsed" desc=" Generated Getter: okCommand20
"> //GEN-BEGIN:|322-getter|0|322-preInit
02277     public Command getOkCommand20() {
02278         if (okCommand20 == null) { //GEN-END:|322-getter|0|322-preInit
02279             // write pre-init user code here
02280             okCommand20 = new Command("Ok", Command.OK, 0); //GEN-LINE:|322-getter
|1|322-postInit
02281             // write post-init user code here
02282         } //GEN-BEGIN:|322-getter|2|
02283         return okCommand20;
02284     }
02285     //</editor-fold> //GEN-END:|322-getter|2|
02286
02287
02288
02289     //<editor-fold defaultstate="collapsed" desc=" Generated Getter: waitScreen4
"> //GEN-BEGIN:|327-getter|0|327-preInit

```

```

02294     public WaitScreen getWaitScreen4() {
02295         if (waitScreen4 == null) { //GEN-END:|327-getter|0|327-preInit
02296             // write pre-init user code here
02297             waitScreen4 = new WaitScreen(getDisplay()); //GEN-BEGIN:|327-getter|1|
327-postInit
02298             waitScreen4.setTitle("waitScreen4");
02299             waitScreen4.setCommandListener(this);
02300             waitScreen4.setTask(getTask5()); //GEN-END:|327-getter|1|327-postInit
02301             // write post-init user code here
02302
02303             } //GEN-BEGIN:|327-getter|2|
02304             return waitScreen4;
02305         }
02306         //</editor-fold> //GEN-END:|327-getter|2|
02307
02308         //<editor-fold defaultstate="collapsed" desc=" Generated Getter: task5 "> //GE
N-BEGIN:|330-getter|0|330-preInit
02313     public SimpleCancellableTask getTask5() {
02314         if (task5 == null) { //GEN-END:|330-getter|0|330-preInit
02315             // write pre-init user code here
02316             task5 = new SimpleCancellableTask(); //GEN-BEGIN:|330-getter|1|330-exe
cute
02317             task5.setExecutable(new org.netbeans.microedition.util.Executable() {
02318
30-execute
02319                 // write task-execution user code here
02320                 realizarTransferencia2();
02321                 } //GEN-BEGIN:|330-getter|2|330-postInit
02322             }); //GEN-END:|330-getter|2|330-postInit
02323             // write post-init user code here
02324             } //GEN-BEGIN:|330-getter|3|
02325             return task5;
02326         }
02327         //</editor-fold> //GEN-END:|330-getter|3|
02328
02329         //<editor-fold defaultstate="collapsed" desc=" Generated Getter: cancelComman
d "> //GEN-BEGIN:|337-getter|0|337-preInit
02334     public Command getCancelCommand() {
02335         if (cancelCommand == null) { //GEN-END:|337-getter|0|337-preInit
02336             // write pre-init user code here
02337             cancelCommand = new Command("Cancel", Command.CANCEL, 0); //GEN-LINE:|
337-getter|1|337-postInit
02338             // write post-init user code here
02339             } //GEN-BEGIN:|337-getter|2|
02340             return cancelCommand;
02341         }
02342         //</editor-fold> //GEN-END:|337-getter|2|
02343
02344         //<editor-fold defaultstate="collapsed" desc=" Generated Getter: splashScreen
"> //GEN-BEGIN:|340-getter|0|340-preInit
02349     public SplashScreen getSplashScreen() {
02350         if (splashScreen == null) { //GEN-END:|340-getter|0|340-preInit
02351             // write pre-init user code here
02352             splashScreen = new SplashScreen(getDisplay()); //GEN-BEGIN:|340-getter
|1|340-postInit
02353             splashScreen.setTitle("Acerca de ...");

```



```

02354         splashScreen.setCommandListener(this);
02355         splashScreen.setImage(getImage1());
02356         splashScreen.setTimeout(12000); //GEN-END:|340-getter|1|340-postInit
02357         splashScreen.setTicker(getTickerSalida());
02358         // write post-init user code here
02359     } //GEN-BEGIN:|340-getter|2|
02360     return splashScreen;
02361 }
02362 //</editor-fold> //GEN-END:|340-getter|2|
02363
02364 //<editor-fold defaultstate="collapsed" desc=" Generated Getter: exitCommand2
"> //GEN-BEGIN:|342-getter|0|342-preInit
02369     public Command getExitCommand2() {
02370         if (exitCommand2 == null) //GEN-END:|342-getter|0|342-preInit
02371             // write pre-init user code here
02372             exitCommand2 = new Command("Exit", Command.EXIT, 0); //GEN-LINE:|342-g
etter|1|342-postInit
02373             // write post-init user code here
02374         } //GEN-BEGIN:|342-getter|2|
02375         return exitCommand2;
02376     }
02377 //</editor-fold> //GEN-END:|342-getter|2|
02378
02379 //<editor-fold defaultstate="collapsed" desc=" Generated Getter: image1 "> //G
EN-BEGIN:|346-getter|0|346-preInit
02384     public Image getImage1() {
02385         if (image1 == null) //GEN-END:|346-getter|0|346-preInit
02386             // write pre-init user code here
02387             try //GEN-BEGIN:|346-getter|1|346-@java.io.IOException
02388                 image1 = Image.createImage("/java_powered_logo.png");
02389             } catch (java.io.IOException e) //GEN-END:|346-getter|1|346-@java.io
.IOException
02390                 e.printStackTrace();
02391             } //GEN-LINE:|346-getter|2|346-postInit
02392             // write post-init user code here
02393         } //GEN-BEGIN:|346-getter|3|
02394         return image1;
02395     }
02396 //</editor-fold> //GEN-END:|346-getter|3|
02397
02398 //<editor-fold defaultstate="collapsed" desc=" Generated Getter: cancelComman
d1 "> //GEN-BEGIN:|349-getter|0|349-preInit
02403     public Command getCancelCommand1() {
02404         if (cancelCommand1 == null) //GEN-END:|349-getter|0|349-preInit
02405             // write pre-init user code here
02406             cancelCommand1 = new Command("Cancel", Command.CANCEL, 0); //GEN-LINE:
|349-getter|1|349-postInit
02407             // write post-init user code here
02408         } //GEN-BEGIN:|349-getter|2|
02409         return cancelCommand1;
02410     }
02411 //</editor-fold> //GEN-END:|349-getter|2|
02412
02413 //<editor-fold defaultstate="collapsed" desc=" Generated Getter: okCommand21
"> //GEN-BEGIN:|358-getter|0|358-preInit
02418     public Command getOkCommand21() {
02419         if (okCommand21 == null) //GEN-END:|358-getter|0|358-preInit

```

```

02420         // write pre-init user code here
02421         okCommand21 = new Command("Ok", Command.OK, 0); //GEN-LINE:|358-getter
|1|358-postInit
02422         // write post-init user code here
02423         } //GEN-BEGIN:|358-getter|2|
02424         return okCommand21;
02425     }
02426     //</editor-fold> //GEN-END:|358-getter|2|
02427
02428     //<editor-fold defaultstate="collapsed" desc=" Generated Getter: form4 "> //GE
N-BEGIN:|352-getter|0|352-preInit
02433     public Form getForm4() {
02434         if (form4 == null) { //GEN-END:|352-getter|0|352-preInit
02435             // write pre-init user code here
02436             form4 = new Form("Carga", new Item[] { getChoiceGroup3() }); //GEN-BEG
IN:|352-getter|1|352-postInit
02437             form4.addCommand(getItemCommand4());
02438             form4.addCommand(getItemCommand5());
02439             form4.addCommand(getOkCommand21());
02440             form4.setCommandListener(this); //GEN-END:|352-getter|1|352-postInit
02441             // write post-init user code here
02442             } //GEN-BEGIN:|352-getter|2|
02443             return form4;
02444         }
02445         //</editor-fold> //GEN-END:|352-getter|2|
02446
02447     //<editor-fold defaultstate="collapsed" desc=" Generated Getter: waitScreen5
"> //GEN-BEGIN:|354-getter|0|354-preInit
02452     public WaitScreen getWaitScreen5() {
02453         if (waitScreen5 == null) { //GEN-END:|354-getter|0|354-preInit
02454             // write pre-init user code here
02455             waitScreen5 = new WaitScreen(getDisplay()); //GEN-BEGIN:|354-getter|1|
354-postInit
02456             waitScreen5.setTitle("waitScreen5");
02457             waitScreen5.setCommandListener(this);
02458             waitScreen5.setTask(getTask6()); //GEN-END:|354-getter|1|354-postInit
02459             // write post-init user code here
02460             } //GEN-BEGIN:|354-getter|2|
02461             return waitScreen5;
02462         }
02463         //</editor-fold> //GEN-END:|354-getter|2|
02464
02465     //<editor-fold defaultstate="collapsed" desc=" Generated Getter: task6 "> //GE
N-BEGIN:|357-getter|0|357-preInit
02470     public SimpleCancellableTask getTask6() {
02471         if (task6 == null) { //GEN-END:|357-getter|0|357-preInit
02472             // write pre-init user code here
02473             task6 = new SimpleCancellableTask(); //GEN-BEGIN:|357-getter|1|357-exe
cute
02474             task6.setExecutable(new org.netbeans.microedition.util.Executable() {
02475                 public void execute() throws Exception { //GEN-END:|357-getter|1|3
57-execute
02476                     // write task-execution user code here
02477                     realizarTransferencia3();
02478                     } //GEN-BEGIN:|357-getter|2|357-postInit
02479             }); //GEN-END:|357-getter|2|357-postInit

```

```

02480         // write post-init user code here
02481     }//GEN-BEGIN:|357-getter|3|
02482     return task6;
02483 }
02484 //</editor-fold>//GEN-END:|357-getter|3|
02485
02486 //<editor-fold defaultstate="collapsed" desc=" Generated Getter: choiceGroup3
">//GEN-BEGIN:|362-getter|0|362-preInit
02491 public ChoiceGroup getChoiceGroup3() {
02492     if (choiceGroup3 == null) { //GEN-END:|362-getter|0|362-preInit
02493         // write pre-init user code here
02494
02495
02496         String tmpAgrega = "";
02497         int r = 0;
02498         int size = 0;
02499         size = salidaSeleccion2.size();
02500         boolean[] d = new boolean[size];
02501
02502         choiceGroup3 = new ChoiceGroup("Selecciona los archivos a sincronizar
", Choice.MULTIPLE); //GEN-LINE:|362-getter|1|362-postInit
02503
02504         while(!salidaSeleccion2.isEmpty()){
02505
02506             tmpAgrega = salidaSeleccion2.firstElement().toString();
02507             choiceGroup3.append(tmpAgrega.toString(), null);
02508             salidaSeleccion2.removeElementAt(r);
02509         }
02510
02511         for(int in = 0; in < size; in++){
02512             d[in]=false;
02513         }
02514
02515         choiceGroup3.setSelectedFlags(d);
02516
02517         // write post-init user code here
02518     }//GEN-BEGIN:|362-getter|2|
02519     return choiceGroup3;
02520 }
02521 //</editor-fold>//GEN-END:|362-getter|2|
02522
02523 //<editor-fold defaultstate="collapsed" desc=" Generated Getter: verArchivosM
ovil ">//GEN-BEGIN:|365-getter|0|365-preInit
02528 public Alert getVerArchivosMovil() {
02529     if (verArchivosMovil == null) { //GEN-END:|365-getter|0|365-preInit
02530         // write pre-init user code here
02531         verArchivosMovil = new Alert("Explorador servidor", "A continuaci\u00
F3n se mostrar\u00E9n los archivos del m\u00F3vil", getImage2(), null); //GEN-BEGI
N:|365-getter|1|365-postInit
02532         verArchivosMovil.setTimeout(2000); //GEN-END:|365-getter|1|365-postIni
t
02533
02534         // write post-init user code here
02535     }//GEN-BEGIN:|365-getter|2|
02536     return verArchivosMovil;
02537 }
02538 //</editor-fold>//GEN-END:|365-getter|2|

```



```

02539     //<editor-fold defaultstate="collapsed" desc=" Generated Getter: verArchivosS
ervidor ">//GEN-BEGIN:|367-getter|0|367-preInit
02544     public Alert getVerArchivosServidor() {
02545         if (verArchivosServidor == null) { //GEN-END:|367-getter|0|367-preInit
02546             // write pre-init user code here
02547             verArchivosServidor = new Alert("Explorador m\u00F3vil", "A continuac
i\u00F3n se mostrar\u00E1n los archivos del m\u00F3vil", getImage2(), null); //GEN
-BEGIN:|367-getter|1|367-postInit
02548             verArchivosServidor.setTimeout(2000); //GEN-END:|367-getter|1|367-post
Init
02549             // write post-init user code here
02550         } //GEN-BEGIN:|367-getter|2|
02551         return verArchivosServidor;
02552     }
02553     //</editor-fold> //GEN-END:|367-getter|2|
02554
02555     //<editor-fold defaultstate="collapsed" desc=" Generated Getter: image2 "> //G
EN-BEGIN:|369-getter|0|369-preInit
02560     public Image getImage2() {
02561         if (image2 == null) { //GEN-END:|369-getter|0|369-preInit
02562             // write pre-init user code here
02563             try { //GEN-BEGIN:|369-getter|1|369-@java.io.IOException
02564                 image2 = Image.createImage("/ok.jpg");
02565             } catch (java.io.IOException e) { //GEN-END:|369-getter|1|369-@java.io
.IOException
02566                 e.printStackTrace();
02567             } //GEN-LINE:|369-getter|2|369-postInit
02568             // write post-init user code here
02569         } //GEN-BEGIN:|369-getter|3|
02570         return image2;
02571     }
02572     //</editor-fold> //GEN-END:|369-getter|3|
02573
02574     //<editor-fold defaultstate="collapsed" desc=" Generated Getter: image3 "> //G
EN-BEGIN:|370-getter|0|370-preInit
02579     public Image getImage3() {
02580         if (image3 == null) { //GEN-END:|370-getter|0|370-preInit
02581             // write pre-init user code here
02582             try { //GEN-BEGIN:|370-getter|1|370-@java.io.IOException
02583                 image3 = Image.createImage("/error.png");
02584             } catch (java.io.IOException e) { //GEN-END:|370-getter|1|370-@java.io
.IOException
02585                 e.printStackTrace();
02586             } //GEN-LINE:|370-getter|2|370-postInit
02587             // write post-init user code here
02588         } //GEN-BEGIN:|370-getter|3|
02589         return image3;
02590     }
02591     //</editor-fold> //GEN-END:|370-getter|3|
02592
02593     //<editor-fold defaultstate="collapsed" desc=" Generated Getter: itemCommand2
"> //GEN-BEGIN:|371-getter|0|371-preInit
02598     public Command getItemCommand2() {
02599         if (itemCommand2 == null) { //GEN-END:|371-getter|0|371-preInit
02600             // write pre-init user code here
02601             itemCommand2 = new Command("Todos", Command.ITEM, 0); //GEN-LINE:|371-
getter|1|371-postInit

```

```

02602         // write post-init user code here
02603     }//GEN-BEGIN:|371-getter|2|
02604     return itemCommand2;
02605 }
02606 //</editor-fold>//GEN-END:|371-getter|2|
02607
02608 //<editor-fold defaultstate="collapsed" desc=" Generated Getter: itemCommand3
">//GEN-BEGIN:|373-getter|0|373-preInit
02613     public Command getItemCommand3() {
02614         if (itemCommand3 == null) { //GEN-END:|373-getter|0|373-preInit
02615             // write pre-init user code here
02616             itemCommand3 = new Command("Ninguno", Command.ITEM, 0); //GEN-LINE:|37
3-getter|1|373-postInit
02617             // write post-init user code here
02618         } //GEN-BEGIN:|373-getter|2|
02619         return itemCommand3;
02620     }
02621 //</editor-fold>//GEN-END:|373-getter|2|
02622
02623 //<editor-fold defaultstate="collapsed" desc=" Generated Getter: itemCommand4
">//GEN-BEGIN:|375-getter|0|375-preInit
02628     public Command getItemCommand4() {
02629         if (itemCommand4 == null) { //GEN-END:|375-getter|0|375-preInit
02630             // write pre-init user code here
02631             itemCommand4 = new Command("Todos", Command.ITEM, 0); //GEN-LINE:|375-
getter|1|375-postInit
02632             // write post-init user code here
02633         } //GEN-BEGIN:|375-getter|2|
02634         return itemCommand4;
02635     }
02636 //</editor-fold>//GEN-END:|375-getter|2|
02637
02638 //<editor-fold defaultstate="collapsed" desc=" Generated Getter: itemCommand5
">//GEN-BEGIN:|377-getter|0|377-preInit
02643     public Command getItemCommand5() {
02644         if (itemCommand5 == null) { //GEN-END:|377-getter|0|377-preInit
02645             // write pre-init user code here
02646             itemCommand5 = new Command("Ninguno", Command.ITEM, 0); //GEN-LINE:|37
7-getter|1|377-postInit
02647             // write post-init user code here
02648         } //GEN-BEGIN:|377-getter|2|
02649         return itemCommand5;
02650     }
02651 //</editor-fold>//GEN-END:|377-getter|2|
02652
02653 //<editor-fold defaultstate="collapsed" desc=" Generated Getter: okCommand22
">//GEN-BEGIN:|383-getter|0|383-preInit
02658     public Command getOkCommand22() {
02659         if (okCommand22 == null) { //GEN-END:|383-getter|0|383-preInit
02660             // write pre-init user code here
02661             okCommand22 = new Command("Ok", Command.OK, 0); //GEN-LINE:|383-getter
|1|383-postInit
02662             // write post-init user code here
02663         } //GEN-BEGIN:|383-getter|2|
02664         return okCommand22;
02665     }
02666 //</editor-fold>//GEN-END:|383-getter|2|

```

```

02667
02668     //<editor-fold defaultstate="collapsed" desc=" Generated Getter: backCommand1
0 ">//GEN-BEGIN:|385-getter|0|385-preInit
02673     public Command getBackCommand10() {
02674         if (backCommand10 == null) { //GEN-END:|385-getter|0|385-preInit
02675             // write pre-init user code here
02676             backCommand10 = new Command("Back", Command.BACK, 0); //GEN-LINE:|385-
getter|1|385-postInit
02677             // write post-init user code here
02678         } //GEN-BEGIN:|385-getter|2|
02679         return backCommand10;
02680     }
02681     //</editor-fold>//GEN-END:|385-getter|2|
02682
02683     //<editor-fold defaultstate="collapsed" desc=" Generated Getter: form5 ">//GE
N-BEGIN:|380-getter|0|380-preInit
02688     public Form getForm5() {
02689         if (form5 == null) { //GEN-END:|380-getter|0|380-preInit
02690             // write pre-init user code here
02691             form5 = new Form("Opciones de Sincronizaci\u00F3n", new Item[] {
getChoiceGroup4() }); //GEN-BEGIN:|380-getter|1|380-postInit
02692             form5.addCommand(getOkCommand22());
02693             form5.addCommand(getBackCommand10());
02694             form5.setCommandListener(this); //GEN-END:|380-getter|1|380-postInit
02695             // write post-init user code here
02696         } //GEN-BEGIN:|380-getter|2|
02697         return form5;
02698     }
02699     //</editor-fold>//GEN-END:|380-getter|2|
02700
02701     //<editor-fold defaultstate="collapsed" desc=" Generated Getter: choiceGroup4
">//GEN-BEGIN:|381-getter|0|381-preInit
02706     public ChoiceGroup getChoiceGroup4() {
02707         if (choiceGroup4 == null) { //GEN-END:|381-getter|0|381-preInit
02708             // write pre-init user code here
02709             choiceGroup4 = new ChoiceGroup("", Choice.EXCLUSIVE); //GEN-BEGIN:|381-
-getter|1|381-postInit
02710             choiceGroup4.append("Sincronizaci\u00F3n sin reemplazo", null);
02711             choiceGroup4.append("Sincronizaci\u00F3n con reemplazo", null);
02712             choiceGroup4.append("Sincronizaci\u00F3n con reemplazo y preguntar en
caso de reemplazo", null);
02713             choiceGroup4.setSelectedFlags(new boolean[] { true, false, false }); //
/GEN-END:|381-getter|1|381-postInit
02714             // write post-init user code here
02715         }
02716         else {
02717             choiceGroup4.setSelectedFlags(new boolean[] { true, false, false });
02718         } //GEN-BEGIN:|381-getter|2|
02719         return choiceGroup4;
02720     }
02721     //</editor-fold>//GEN-END:|381-getter|2|
02722
02723
02724
02725     //<editor-fold defaultstate="collapsed" desc=" Generated Getter: waitScreen6
">//GEN-BEGIN:|410-getter|0|410-preInit
02730     public WaitScreen getWaitScreen6() {

```



```

02731         if (waitScreen6 == null) { //GEN-END:|410-getter|0|410-preInit
02732             // write pre-init user code here
02733             waitScreen6 = new WaitScreen(getDisplay()); //GEN-BEGIN:|410-getter|1|
410-postInit
02734             waitScreen6.setTitle("waitScreen6");
02735             waitScreen6.setCommandListener(this);
02736             waitScreen6.setTask(getTask7()); //GEN-END:|410-getter|1|410-postInit
02737             // write post-init user code here
02738         } //GEN-BEGIN:|410-getter|2|
02739         return waitScreen6;
02740     }
02741 } //</editor-fold> //GEN-END:|410-getter|2|
02742
02743 //<editor-fold defaultstate="collapsed" desc=" Generated Getter: task7 " > //GE
N-BEGIN:|413-getter|0|413-preInit
02748     public SimpleCancellableTask getTask7() {
02749         if (task7 == null) { //GEN-END:|413-getter|0|413-preInit
02750             // write pre-init user code here
02751             task7 = new SimpleCancellableTask(); //GEN-BEGIN:|413-getter|1|413-exe
cute
02752             task7.setExecutable(new org.netbeans.microedition.util.Executable() {
02753
02754                 public void execute() throws Exception { //GEN-END:|413-getter|1|4
13-execute
02754                     // write task-execution user code here
02755                     if (selSincGeneral.equals("Sincronización sin reemplazo")) {
02756                         realizarSincronizacion1();
02757
02758                         //se crea la bitacora
02759                         creacionBitacora();
02760                         //se crea la comparación de la bitacora en modo descarga
02761                         //y se crea el archivo transfer
02762                         comparaBitacorasDown("bitacora.xml");
02763                         //se realiza la lectura de transfer, se descargan los arc
02764                     }
02765                     hivos
02766                     //y se crean los nuevos archivos
02767                     realizarTransferenciaDescargaAIO();
02768
02769                     //ya no es necesario volver a crear la bitacora
02770                     //se crea la comparación de la bitacora en modo carga
02771                     comparaBitacorasUp("bitacora.xml");
02772                     //se realiza la lectura de transfer y se envian los archi
02773                     vos
02774                     realizarTransferenciaCargaAIO();
02775                 }
02776                 else if (selSincGeneral.equals("Sincronización con reemplazo"))
02777             ){
02778                 realizarSincronizacion2();
02779
02780                 //Se verifica si se ha seleccionado Reemplazo desde el
02781                 //cliente o reemplazo desde el servidor
02782                 if (selSincReemplazo.equals("Reemplazo desde el Cliente"))
02783             {
02784                     System.out.println("Cliente!!!!!!!!!!!!!!");
02785
02786                     //se crea la bitacora

```

```

02785         creacionBitacora();
02786
02788         //El cliente(móvil) envía todos sus archivos al servi
dor
02789         //Se crea la comparación de la bitacora en modo carga
02790
02791         comparaBitacorasUp("bitacora.xml");
02792         //se realiza la lectura de transfer y se envian los a
rchivos
02793         realizarTransferenciaCargaAIO();
02794         //Se descargan solo los archivos del servidor que no
tiene
02795         //el cliente(móvil)
02796
02797         //se cambia el modo de la descarga para reutilizar los
metodos
02798         selSincGeneral = "Sincronización sin reemplazo";
02799         //Se crea el archivo transfer
02800         comparaBitacorasDown("bitacora.xml");
02801         //se realiza la lectura de transfer, se descargan los
archivos
02802         //y se crean los nuevos archivos
02803         realizarTransferenciaDescargaAIO();
02804
02805     }
02806     else if(selSincReemplazo.equals("Reemplazo desde el Servi
dor")) {
02807         System.out.println("Servidor!!!!!!!!!!!!!!");
02808
02810
02811         //se crea la bitacora
02812         creacionBitacora();
02813
02814         //Se crea el archivo transfer
02815         comparaBitacorasDown("bitacora.xml");
02816         //se realiza la lectura de transfer, se descargan los
archivos
02817         //y se crean los nuevos archivos
02818         realizarTransferenciaDescargaAIO();
02819
02820         //se cambia el modo de la descarga para reutilizar los
metodos
02821         selSincGeneral = "Sincronización sin reemplazo";
02822
02824
02825         //Se crea el archivo transfer
02826         comparaBitacorasUp("bitacora.xml");
02827         //se realiza la lectura de transfer y se envian los a
rchivos
02828         realizarTransferenciaCargaAIO();
02829
02830     }
02831
02832 }
02833
02834     else if(selSincGeneral.equals("Sincronización con reemplazo y

```

```

preguntar en caso de reemplazo)){
02835         realizarSincronizacion3();
02836
02837         //se crea la bitacora
02838         creacionBitacora();
02839
02840         //Se crea el archivo transfer
02841         comparaBitacorasDown("bitacora.xml");
02842         //se realiza la lectura de transfer, se descargan los arc
02843
hivos
02844         //y se crean los nuevos archivos
02845         realizarTransferenciaDescargaCargaAIODS();
02846
02847         //se cambia el modo de la descarga para reutilizar los met
odos
02848         selSincGeneral = "Sincronización sin reemplazo";
02849
02850         //Envía los archivos que no tiene el servidor del cliente
(móvil)
02851
02852         //ya no es necesario volver a crear la bitacora
02853         //se crea la comparación de la bitacora en modo carga
02854         comparaBitacorasUp("bitacora.xml");
02855         //se realiza la lectura de transfer y se envían los archi
vos
02856         realizarTransferenciaCargaAIO();
02857
02858     }
02859
02860     }//GEN-BEGIN:|413-getter|2|413-postInit
02861     });//GEN-END:|413-getter|2|413-postInit
02862     // write post-init user code here
02863     }//GEN-BEGIN:|413-getter|3|
02864     return task7;
02865 }
02866 //</editor-fold>//GEN-END:|413-getter|3|
02867
02868 //<editor-fold defaultstate="collapsed" desc=" Generated Getter: okCommand23
">//GEN-BEGIN:|419-getter|0|419-preInit
02874     public Command getOkCommand23() {
02875         if (okCommand23 == null) { //GEN-END:|419-getter|0|419-preInit
02876             // write pre-init user code here
02877             okCommand23 = new Command("Ok", Command.OK, 0); //GEN-LINE:|419-getter
|1|419-postInit
02878             // write post-init user code here
02879             } //GEN-BEGIN:|419-getter|2|
02880             return okCommand23;
02881         }
02882     } //GEN-END:|419-getter|2|
02883
02884     //<editor-fold defaultstate="collapsed" desc=" Generated Getter: form6 ">//GE
N-BEGIN:|417-getter|0|417-preInit
02889     public Form getForm6() {
02890         if (form6 == null) { //GEN-END:|417-getter|0|417-preInit
02891             // write pre-init user code here
02892             form6 = new Form("Selecciona las opciones extra de sincronizaci\u00F3n", new Item[] { getChoiceGroup5() }); //GEN-BEGIN:|417-getter|1|417-postInit

```



```

02893         form6.addCommand(getOkCommand23());
02894         form6.setCommandListener(this); //GEN-END:|417-getter|1|417-postInit
02895         // write post-init user code here
02896     } //GEN-BEGIN:|417-getter|2|
02897     return form6;
02898 }
02899 //</editor-fold> //GEN-END:|417-getter|2|
02900
02901 //<editor-fold defaultstate="collapsed" desc=" Generated Getter: choiceGroup5
"> //GEN-BEGIN:|418-getter|0|418-preInit
02906 public ChoiceGroup getChoiceGroup5() {
02907     if (choiceGroup5 == null) { //GEN-END:|418-getter|0|418-preInit
02908         // write pre-init user code here
02909         choiceGroup5 = new ChoiceGroup("", Choice.EXCLUSIVE); //GEN-BEGIN:|418
-getter|1|418-postInit
02910         choiceGroup5.append("Reemplazo desde el Cliente", null);
02911         choiceGroup5.append("Reemplazo desde el Servidor", null);
02912         choiceGroup5.setSelectedFlags(new boolean[] { true, false }); //GEN-EN
D:|418-getter|1|418-postInit
02913
02914         // write post-init user code here
02915     } //GEN-BEGIN:|418-getter|2|
02916     return choiceGroup5;
02917 }
02918 //</editor-fold> //GEN-END:|418-getter|2|
02919
02920 //<editor-fold defaultstate="collapsed" desc=" Generated Method: method "> //G
EN-BEGIN:|422-if|0|422-preIf
02924 public void method() { //GEN-END:|422-if|0|422-preIf
02925     // enter pre-if user code here
02926
02927     if (choiceGroup4.getSelectedIndex() == 0) { //GEN-LINE:|422-if|1|423-preAc
tion
02928         // write pre-action user code here
02929         switchDisplayable(null, getWaitScreen6()); //GEN-LINE:|422-if|2|423-po
stAction
02930
02931         // write post-action user code here
02932     } else { //GEN-LINE:|422-if|3|424-preAction
02933         // write pre-action user code here
02934         method1(); //GEN-LINE:|422-if|4|424-postAction
02935         // write post-action user code here
02936     } //GEN-LINE:|422-if|5|422-postIf
02937     // enter post-if user code here
02938 } //GEN-BEGIN:|422-if|6|
02939 //</editor-fold> //GEN-END:|422-if|6|
02940
02941 //<editor-fold defaultstate="collapsed" desc=" Generated Method: method1 "> //
GEN-BEGIN:|425-if|0|425-preIf
02944 public void method1() { //GEN-END:|425-if|0|425-preIf
02945     // enter pre-if user code here
02946
02947     if (choiceGroup4.getSelectedIndex() == 1) { //GEN-LINE:|425-if|1|426-preAc
tion
02948         // write pre-action user code here
02949         switchDisplayable(null, getForm6()); //GEN-LINE:|425-if|2|426-postActi
on
02950         // write post-action user code here

```

```

02951         } else { //GEN-LINE:|425-if|3|427-preAction
02952             // write pre-action user code here
02953             switchDisplayable(null, getWaitScreen6()); //GEN-LINE:|425-if|4|427-po
stAction
02954             // write post-action user code here
02955         } //GEN-LINE:|425-if|5|425-postIf
02956         // enter post-if user code here
02957     } //GEN-BEGIN:|425-if|6|
02958     //</editor-fold> //GEN-END:|425-if|6|
02959
02960
02961
02962     //<editor-fold defaultstate="collapsed" desc=" Generated Getter: okCommand24
"> //GEN-BEGIN:|442-getter|0|442-preInit
02967     public Command getOkCommand24() {
02968         if (okCommand24 == null) { //GEN-END:|442-getter|0|442-preInit
02969             // write pre-init user code here
02970             okCommand24 = new Command("Ok", Command.OK, 0); //GEN-LINE:|442-getter
|1|442-postInit
02971             // write post-init user code here
02972         } //GEN-BEGIN:|442-getter|2|
02973         return okCommand24;
02974     }
02975     //</editor-fold> //GEN-END:|442-getter|2|
02976
02977     //<editor-fold defaultstate="collapsed" desc=" Generated Getter: textBox "> //
GEN-BEGIN:|440-getter|0|440-preInit
02982     public TextBox getTextBox() {
02983         if (textBox == null) { //GEN-END:|440-getter|0|440-preInit
02984             // write pre-init user code here
02985
02986             textBox = new TextBox("Inicio de sincronizaci\u00F3n", "\n\nHas sel
eccionado :\n" + choiceGroup4.getString(choiceGroup4.getSelectedIndex()) + "\n\nP
resiona OK para continuar ...", 200, TextField.ANY | TextField.UNEDITABLE); //GEN-
BEGIN:|440-getter|1|440-postInit
02987             textBox.addCommand(getOkCommand24());
02988             textBox.addCommand(getCancelCommand2());
02989             textBox.setCommandListener(this); //GEN-END:|440-getter|1|440-postInit
02990
02991             // write post-init user code here
02992         } else {
02993             textBox.setString("\n\nHas seleccionado :\n" + choiceGroup4.getStri
ng(choiceGroup4.getSelectedIndex()) + "\n\nPresiona OK para continuar ...");
02994         } //GEN-BEGIN:|440-getter|2|
02995         return textBox;
02996     }
02997     //</editor-fold> //GEN-END:|440-getter|2|
02998
02999     //<editor-fold defaultstate="collapsed" desc=" Generated Getter: cancelComman
d2 "> //GEN-BEGIN:|445-getter|0|445-preInit
03003     public Command getCancelCommand2() {
03004         if (cancelCommand2 == null) { //GEN-END:|445-getter|0|445-preInit
03005             // write pre-init user code here
03006             cancelCommand2 = new Command("Cancel", Command.CANCEL, 0); //GEN-LINE:
|445-getter|1|445-postInit
03007             // write post-init user code here
03008         } //GEN-BEGIN:|445-getter|2|

```

```
03009         return cancelCommand2;
03010     }
03011     //</editor-fold>//GEN-END:|445-getter|2|
03012
03013     //<editor-fold defaultstate="collapsed" desc=" Generated Getter: sincCorrecto
">//GEN-BEGIN:|449-getter|0|449-preInit
03018     public Alert getSincCorrecto() {
03019         if (sincCorrecto == null) { //GEN-END:|449-getter|0|449-preInit
03020             // write pre-init user code here
03021             sincCorrecto = new Alert("Sincronizaci\u00F3n correcta", "La sincroni
zaci\u00F3n se ha realizado correctamente", getImage2(), null); //GEN-BEGIN:|449-g
etter|1|449-postInit
03022             sincCorrecto.setTimeout(Alert.FOREVER); //GEN-END:|449-getter|1|449-po
stInit
03023             // write post-init user code here
03024         } //GEN-BEGIN:|449-getter|2|
03025         return sincCorrecto;
03026     }
03027     //</editor-fold>//GEN-END:|449-getter|2|
03028
03029     //<editor-fold defaultstate="collapsed" desc=" Generated Getter: SincError ">
//GEN-BEGIN:|450-getter|0|450-preInit
03034     public Alert getSincError() {
03035         if (SincError == null) { //GEN-END:|450-getter|0|450-preInit
03036             // write pre-init user code here
03037             SincError = new Alert("Sincronizaci\u00F3n incorrecta", "La sincroniz
aci\u00F3n no se ha realizado correctamente", getImage3(), null); //GEN-BEGIN:|450
-getter|1|450-postInit
03038             SincError.setTimeout(Alert.FOREVER); //GEN-END:|450-getter|1|450-postI
nit
03039             // write post-init user code here
03040         } //GEN-BEGIN:|450-getter|2|
03041         return SincError;
03042     }
03043     //</editor-fold>//GEN-END:|450-getter|2|
03044
03045
03046
03047
03048
03049
03050
03051
03052
03053
03054
03055
03056
03057
03058
03059
03060
03061
03062
03063
03064
03065
```



```
03066
03067
03072 public Display getDisplay () {
03073     return Display.getDisplay(this);
03074 }
03075
03076
03080 public void exitMIDlet() {
03081     switchDisplayable (null, null);
03082     destroyApp(true);
03083     notifyDestroyed();
03084 }
03085
03090 public void startApp() {
03091     if (midletPaused) {
03092         resumeMIDlet ();
03093     } else {
03094         initialize ();
03095         startMIDlet ();
03096     }
03097     midletPaused = false;
03098 }
03099
03103 public void pauseApp() {
03104     midletPaused = true;
03105 }
03106
03111 public void destroyApp(boolean unconditional) {
03112 }
03113
03114
03115 public Ticker getTicker() {
03116     if (ticker == null) {
03117         // write pre-init user code here
03118         ticker = new Ticker("");
03119         // write post-init user code here
03120         //#ifdef LoginScreenServletExample
03121         //#
03122         ticker.setString("The sample is in online mode. Use username/pass
word john/peanuts for successfull connection.");
03123         //#else
03124         ticker.setString("Pantalla de inicio de sesion: Ingresa tu usuario y
password.");
03125         //#endif
03126     }
03127     return ticker;
03128 }
03129
03130
03131 public Ticker getTickerSalida() {
03132     if (tickerSalida == null) {
03133         // write pre-init user code here
03134         tickerSalida = new Ticker("");
03135         // write post-init user code here
03136         tickerSalida.setString("Proyecto Terminal - " +
03137             "Asesor: Beatriz Adriana Gonzalez Beltran - " +
03138             "Autor: Rodrigo Ramirez Reyes - " +
```

```
03139         "Universidad Autonoma Metropolitana UAM Azcapotzalco");
03140     }
03141     return tickerSalida;
03142 }
03143
03144 private void realizarSincronizacion1() throws Exception
03145 {
03146     System.out.println("Elegiste la : - opcion 1");
03147
03148     //Todo bien
03149     if(true)
03150         return;
03151
03152     throw new Exception("Algo salió mal");
03153 }
03154
03155 private void realizarSincronizacion2() throws Exception
03156 {
03157     System.out.println("wait: - opcion 2");
03158
03159     //Todo bien
03160     if(true)
03161         return;
03162
03163     throw new Exception("Algo salió mal");
03164 }
03165
03166 private void realizarSincronizacion3() throws Exception
03167 {
03168     System.out.println("wait: - opcion 3");
03169     //Todo bien
03170     if(true)
03171         return;
03172
03173     throw new Exception("Algo salió mal");
03174 }
03175
03176 private Vector realizarTransferencial() throws Exception
03177 {
03178     System.out.println("Entro a la transferencia");
03179
03180     //abre el archivo de transferencia
03181
03182     String archivo_trans = "transfer.xml";
03183
03184     byte r[] = new byte[1024];
03185     byte t[] = new byte[1024];
03186     String salida = null;
03187     String archivos = "";
03188     String archivo = "";
03189     String contenido = "";
03190     String rutaServicioWeb = "";
```

```

03196
03197         try {
03198             //se abre el archivo de transferencia
03199
03200             FileConnection fcTrans = (FileConnection)
03201                 Connector.open("file:///root1/" + archivo_trans);
03202             if(!fcTrans.exists()) {
03203                 throw new IOException("File does not exist");
03204             }
03205
03206             //se obtienen los datos
03207
03208             InputStream isTrans = fcTrans.openInputStream();//fcTrans.o
penInputStream();
03209
03210
03211
03212             //Con ayuda del StringBuffer, se almacenaran los datos en un
vector
03213             StringBuffer bufArchivo = new StringBuffer();
03214             Vector lineas = new Vector();
03215
03216             int c ;
03217
03218             //Se obtienen por lines, que en este caso implican un archivo
03219
03220             while ((c = isTrans.read()) != -1)
03221             {
03222                 char ch = (char)c;
03223                 if (ch == '\n')
03224                 {
03225                     lineas.addElement(bufArchivo.toString());
03226                     bufArchivo.delete(0,bufArchivo.length());
03227                 }
03228                 else
03229                     bufArchivo.append(ch);
03230             }
03231
03232             int d = 0;
03233             int ind = 0;
03234             String tmpLineas="";
03235
03236
03237             System.out.println("El vector tiene : " + lineas.size() + "
lineas");
03238
03239             while(!lineas.isEmpty()){
03240
03241
03242                 System.out.println("Archivo No. " + ind + " - " + linea
s.firstElement().toString());
03243
03244                 //Aqui se obtendrán los datos y se realizarán los nuevo
s archivos
03245
03246                 tmpLineas = lineas.firstElement().toString();

```

```
03247 //
03248 // //Creamos una condicional para saber si se trata de un
    directorio
03249 // //y en caso de que si, eliminar las etiquetas
03250 //
03251 //
03252 // tmpLineas = tmpLineas.substring(1, tmpLineas.length()-1
    );
03253 //
03254 // //Aqui se lanzará un form con los archivos en forma de
03255 // //checkboxlist para volver a crear el vector solamente con

03256 // //los archivos que se requieren descargar
03257
03258
03259
03260 //
03261 // System.out.println("El archivo es : " + tmpLineas);
03262 //
03263 // if(!tmpLineas.endsWith("/"))
03264 // {
03265 //
03266 // //creamos la ruta con la información que se va a en
    viar
03267 // rutaServicioWeb = "http://localhost:8084/XMLTest/re
    sources/archivoport/descarga/?nombrearchivo=" + tmpLineas;
03268 //
03269 // //recibimos la información del WS una vez que compa
    ró la bitacora local enviada
03270 // //con la bitocora del servidor
03271 // contenido = Parser.ConectorWeb.sendHttpGet(rutaServ
    icioWeb);
03272 //
03273 // System.out.println("Lo que recibió de archivo:" + c
    ontenido + " :");
03274 //
03275 //
03276 // //paso la información recibida a un byte[]
03277 //
03278 // t = contenido.getBytes();
03279 //
03280 // //Creo el archivo que será el que me ayudará con la
    transferencia
03281 //
03282 // archivo(t, tmpLineas);
03283 //
03284 // System.out.println("Se ha creado el archivo : " + t
    mpLineas);
03285 //
03286 // }
03287
03288 //
03289 // lineas.removeElementAt(d);
03290 // ind++;
03291 // }
03292
03293
```

```
03294         fcTrans.close();
03295         isTrans.close();
03296
03297         //Todo bien
03298         return lineas;
03299
03300     }catch (IOException s){System.out.println(s);}
03301
03302     throw new Exception("No se creo la bitacora remota");
03303 }
03304
03305 private void realizarTransferenciaDescargaAIO() throws Exception
03306 {
03307
03308     System.out.println("Entro a la transferencia");
03309
03310     //abre el archivo de transferencia
03311
03312     String archivo_trans = "transfer.xml";
03313
03314     byte r[] = new byte[1024];
03315     byte t[] = new byte[1024];
03316     String salida = null;
03317     String archivos = "";
03318     String archivo = "";
03319     String contenido = "";
03320     String rutaServicioWeb = "";
03321
03322     try {
03323         //se abre el archivo de transferencia
03324
03325         FileConnection fcTrans = (FileConnection)
03326             Connector.open("file:///root1/" + archivo_trans);
03327         if(!fcTrans.exists()) {
03328             throw new IOException("File does not exist");
03329         }
03330
03331         //se obtienen los datos
03332
03333         InputStream isTrans = fcTrans.openInputStream();//fcTrans.o
03334 penInputStream();
03335
03336
03337         //Con ayuda del StringBuffer, se almacenaran los datos en un
03338 vector
03339         StringBuffer bufArchivo = new StringBuffer();
03340         Vector lineas = new Vector();
03341
03342         int c ;
03343
03344         //Se obtienen por lines, que en este caso implican un archivo
03345
03346         while ((c = isTrans.read()) != -1)
03347         {
03348             char ch = (char)c;
```



```

03348         if (ch == '\n')
03349         {
03350             lineas.addElement(bufArchivo.toString());
03351             bufArchivo.delete(0,bufArchivo.length());
03352         }
03353         else
03354             bufArchivo.append(ch);
03355     }
03356
03357     int d = 0;
03358     int ind = 0;
03359     String tmpLineas="";
03360
03361
03362     System.out.println("El vector tiene : " + lineas.size() + "
archivos";
03363
03364     while(!lineas.isEmpty()){
03365
03366
03367         System.out.println("Archivo No. " + ind + " - " + lineas.
firstElement().toString());
03368
03369         //Aqui se obtendrán los datos y se realizarán los nuevos
archivos
03370
03371         tmpLineas = lineas.firstElement().toString();
03372
03373         //Creamos una condicional para saber si se trata de un di
rectorio
03374         //y en caso de que si, eliminar las etiquetas
03375
03376
03377         tmpLineas = tmpLineas.substring(1, tmpLineas.length()-1);
03378
03379         //Aqui se lanzará un form con los archivos en forma de
//checklist para volver a crear el vector solamente con
//los archivos que se requieren descargar
03380
03381
03382
03383
03384
03385
03386         System.out.println("El archivo es : " + tmpLineas);
03387
03388         if(!tmpLineas.endsWith("/"))
03389         [
03390
03391         //creamos la ruta con la información que se va a envi
ar
03392         rutaServicioWeb = urlDefinida + "/XMLTest/resources/a
rchivoport/descarga/?nombrearchivo=" +
03393             URLEncoder.encode(tmpLineas) + "&usuario=" + usuarioA
ctual +
03394             "&password=" + passwordActual;
03395
03396         //recibimos la información del WS una vez que comparó

```



```
        la bitacora local enviada
03397 //con la bitocora del servidor
03398 contenido = Parser.ConectorWeb.sendHttpGet(rutaServic
ioWeb);
03399
03400 System.out.println("Lo que recibió de archivo:" + con
tenido + ":");
03401
03402
03403 //paso la información recibida a un byte[]
03404
03405 t = contenido.getBytes();
03406
03407 //Creo el archivo que será el que me ayudará con la t
ransferencia
03408
03409 archivo(t, tmpLineas);
03410
03411 System.out.println("Se ha creado el archivo : " + tmp
Lineas);
03412
03413     }
03414
03415
03416     lineas.removeElementAt(d);
03417     ind++;
03418 }
03419
03420
03421     fcTrans.close();
03422     isTrans.close();
03423
03424 //Todo bien
03425 return;
03426
03427     }catch (IOException s){System.out.println(s);}
03428
03429     throw new Exception("No se creo la bitacora remota");
03430 }
03431
03432 private void realizarTransferenciaDescargaCargaAIODS() throws Exception
03433 {
03434
03435     System.out.println("Entro a la transferencia");
03436
03437     //abre el archivo de transferencia
03438
03439     String archivo_trans = "transfer.xml";
03440
03441     byte r[] = new byte[1024];
03442     byte t[] = new byte[1024];
03443     String contenido = "";
03444     String rutaServicioWeb = "";
03445
03446     byte[] w;
03447     String enviaInformacion = "";
03448
```

```

03449         try {
03450             //se abre el archivo de transferencia
03451
03452             FileConnection fcTrans = (FileConnection)
03453                 Connector.open("file:///root1/" + archivo_trans);
03454             if(!fcTrans.exists()) {
03455                 throw new IOException("File does not exist");
03456             }
03457
03458             //se obtienen los datos
03459
03460             InputStream isTrans = fcTrans.openInputStream();//fcTrans.o
03461 penInputStream();
03462
03463
03464             //Con ayuda del StringBuffer, se almacenaran los datos en un
03465 vector
03466             StringBuffer bufArchivo = new StringBuffer();
03467             Vector lineas = new Vector();
03468
03469             int c ;
03470
03471             //Se obtienen por líneas, que en este caso implican un archivo
03472
03473             while ((c = isTrans.read()) != -1)
03474             {
03475                 char ch = (char)c;
03476                 if (ch == '\n')
03477                 {
03478                     lineas.addElement(bufArchivo.toString());
03479                     bufArchivo.delete(0,bufArchivo.length());
03480                 }
03481                 else
03482                     bufArchivo.append(ch);
03483             }
03484
03485             int d = 0;
03486             int ind = 0;
03487             String tmpLineas="";
03488
03489             System.out.println("El vector tiene : " + lineas.size() + "
03490 lineas");
03491
03492             while(!lineas.isEmpty()){
03493
03494                 System.out.println("Archivo No. " + ind + " - " + lineas.
03495 firstElement().toString());
03496
03497                 //Aqui se obtendrán los datos y se realizarán los nuevos
03498 archivos
03499                 tmpLineas = lineas.firstElement().toString();
03499

```

```
03500                                     //Creamos una condicional para saber si se trata de un di
rectorio
03501                                     //y en caso de que si, eliminar las etiquetas
03502
03503
03504                                     tmpLineas = tmpLineas.substring(1, tmpLineas.length()-1);

03505
03506                                     //Aqui se lanzará un form con los archivos en forma de
03507                                     //checklist para volver a crear el vector solamente con
03508                                     //los archivos que se requieren descargar
03509
03510
03511
03512
03513                                     System.out.println("El archivo es : " + tmpLineas);
03514
03515                                     if(!tmpLineas.endsWith("/"))
03516                                     {
03517
03518                                         espera = true;
03519                                         //reemplaza = false;
03520                                         clienteoservidor = "servidor";
03521
03522
03523                                     //copiamos el nombre a la variable nombreAlerta p
ara
03524                                     //enviarla a la Alerta de reemplazo de archivos
03525                                     nombreAlerta = tmpLineas;
03526
03527                                     //Checamos si existe el archivo creando una conex
ion
03528                                     //previamente antes de ir por el contenido
03529
03530                                     System.out.println("checando si : " + tmpLineas +
" existe");
03531
03532                                     FileConnection fcExist = (FileConnection)
03533                                     Connector.open("file:///root1/" + tmpLineas);
03534
03535                                     if(fcExist.exists()) {
03536                                         switchDisplayable(null, getAlertaArchivosCS())
;
03537                                     while(espera)
03538                                     {
03539                                         textBoxAlertaCS.isShown();
03540                                     }
03541                                     System.out.println("En el archivo : " + tmpLin
eas +
" seleccionó " + clienteoservidor);
03542
03543
03544                                     if(clienteoservidor.equals("cliente")){
03545                                         //se realiza la lectura del contenido y se
03546                                         envía el archivo al servidor
03547
03548                                     FileConnection fcSube = (FileConnection)
```

```

03549         Connector.open("file:///root1/" + tmpLineas
);
03550     };
03551     if(!fcSube.exists()) {
03552         throw new IOException("File does not exist");
03553     }
03554
03555     w = new byte[(int)fcSube.fileSize()];
03556     InputStream isSube = fcSube.openInputStream();
03557
03558     isSube.read(w, 0, (int)fcSube.fileSize());
03559     enviaInformacion = w.toString();
03560
03561     //creamos la ruta con la información que se va a enviar
03562     rutaServicioWeb = urlDefinida + "/XMLTest/resources/recibeport/creararchivo/?nombre=" +
03563         URLEncoder.encode(tmpLineas) + "&contenido=" + enviaInformacion.toString() +
03564         "&usuario=" + usuarioActual + "&password=" + passwordActual;
03565
03566     //recibimos la información del WS una vez que subimos la información del archivo
03567     //con la bitocora del servidor
03568     contenido = Parser.ConectorWeb.sendHttpGet(rutaServicioWeb);
03569
03570     System.out.println("Lo que recibió de archivo:" + contenido + ":");
03571
03572     System.out.println("El archivo : " + tmpLineas + " ha sido subido al servidor");
03573
03574     fcSube.close();
03575 }
03576
03577 }
03578
03579 if(clienteoservidor.equals("servidor")){
03580     //aquí poner todo lo que se tiene que hacer si no existe
03581     //que en este caso es descargarlo
03582
03583     //creamos la ruta con la información que se va a enviar
03584     rutaServicioWeb = urlDefinida + "/XMLTest/resources/archivoport/descarga/?nombrearchivo=" +
03585         URLEncoder.encode(tmpLineas) + "&usuario=" + usuarioActual +
03586         "&password=" + passwordActual;
03587
03588     //recibimos la información del WS una vez que com
03589     paró la bitacora local enviada //con la bitocora del servidor

```

```
03590         contenido = Parser.ConectorWeb.sendHttpGet(rutaSe
    rvicioWeb);
03591
03592         System.out.println("Lo que recibió de archivo:" +
    contenido + ":");
03593
03594         //paso la información recibida a un byte[]
03595         t = contenido.getBytes();
03596
03597         //Creo el archivo que será el que me ayudará con
    la transferencia
03600         archivo(t, tmpLineas);
03601
03602         System.out.println("Se ha creado el archivo : " +
    tmpLineas);
03603     }
03604     }
03605     fcExist.close();
03606 }
03607
03608     lines.removeElementAt(d);
03609     ind++;
03610 }
03611
03612     fcTrans.close();
03613     isTrans.close();
03614
03615     //Todo bien
03616     return;
03617 }catch (IOException s){System.out.println(s);}
03618
03619     throw new Exception("No se creo la bitacora remota");
03620 }
03621
03622 private void realizarTransferenciaCargaAIO() throws Exception
03623 {
03624     System.out.println("Entro a la transferencia");
03625
03626     //abre el archivo de transferencia
03627     String archivo_trans = "transfer.xml";
03628
03629     byte r[] = new byte[1024];
03630     byte t[] = new byte[1024];
03631     String salida = null;
03632     String archivos = "";
03633     String archivo = "";
```



```

03643         String contenido = "";
03644         String rutaServicioWeb = "";
03645
03646         byte[] x;
03647         String enviaInformacion = "";
03648
03649         try {
03650             //se abre el archivo de transferencia
03651
03652             FileConnection fcTrans = (FileConnection)
03653                 Connector.open("file://root1/" + archivo_trans);
03654             if(!fcTrans.exists()) {
03655                 throw new IOException("File does not exist");
03656             }
03657
03658             //se obtienen los datos
03659
03660             InputStream isTrans = fcTrans.openInputStream();//fcTrans.o
03661 penInputStream();
03662
03663
03664             //Con ayuda del StringBuffer, se almacenaran los datos en un
03665 vector
03666             StringBuffer bufArchivo = new StringBuffer();
03667             Vector lineas = new Vector();
03668
03669             int c ;
03670
03671             //Se obtienen por líneas, que en este caso implican un archivo
03672
03673             while ((c = isTrans.read()) != -1)
03674             {
03675                 char ch = (char)c;
03676                 if (ch == '\n')
03677                 {
03678                     lineas.addElement(bufArchivo.toString());
03679                     bufArchivo.delete(0,bufArchivo.length());
03680                 }
03681                 else
03682                     bufArchivo.append(ch);
03683             }
03684
03685             int d = 0;
03686             int ind = 0;
03687             String tmpLineas="";
03688
03689             System.out.println("El vector tiene : " + lineas.size() + "
03690 lineas");
03691
03692             while(!lineas.isEmpty()){
03693
03694                 System.out.println("Archivo No. " + ind + " - " + lineas.
03695 firstElement().toString());

```



```
03695
03696 //Aqui se obtendrán los datos y se realizarán los nuevos
archivos
03697 tmpLineas = lineas.firstElement().toString();
03698
03699 //Creamos una condicional para saber si se trata de un di
03700 rectorio
03701 //y en caso de que si, eliminar las etiquetas
03702
03703 tmpLineas = tmpLineas.substring(1, tmpLineas.length()-1);
03704
03705
03706 //Aqui se lanzará un form con los archivos en forma de
03707 //checklist para volver a crear el vector solamente con
03708 //los archivos que se requieren descargar
03709
03710
03711
03712
03713 System.out.println("El archivo es : " + tmpLineas);
03714
03715 if(!tmpLineas.endsWith("/"))
03716 {
03717
03718
03719 FileConnection fcSube = (FileConnection)
03720 Connector.open("file:///root1/" + tmpLineas);
03721 if(!fcSube.exists()) {
03722     throw new IOException("File does not exist");
03723 }
03724
03725
03726 x = new byte[(int)fcSube.fileSize()];
03727 InputStream isSube = fcSube.openInputStream();
03728
03729 isSube.read(x, 0, (int)fcSube.fileSize());
03730 enviaInformacion = x.toString();
03731
03732 //creamos la ruta con la información que se va a enviar
03733
03734 rutaServicioWeb = urlDefinida + "/XMLTest/resources/rec
ibeport/creararchivo/?nombre=" +
03735 URLEncoder.encode(tmpLineas) + "&contenido=" + enviaI
nformacion.toString()+
03736 "&usuario=" + usuarioActual + "&password=" +
passwordActual;
03737 //recibimos la información del WS una vez que subimos
la información del archivo
03738 //con la bitocora del servidor
03739 contenido = Parser.ConectorWeb.sendHttpGet(rutaServic
ioWeb);
03740
03741 System.out.println("Lo que recibió de archivo:" + con
tenido + ":");
```

```
03742
03743         }
03744
03745
03746         lineas.removeElementAt(d);
03747         ind++;
03748     }
03749
03750
03751         fcTrans.close();
03752         isTrans.close();
03753
03754         //Todo bien
03755         return;
03756
03757     }catch(IOException s){System.out.println(s);}
03758
03759     throw new Exception("No se creo la bitacora remota");
03760 }
03761
03762
03763
03764
03765 private Vector seleccionDeArchivos(Vector lineas) throws Exception
03766 {
03767     int d = 0;
03768     int ind = 0;
03769     String tmpLineas="";
03770     Vector imprimir = new Vector();
03771
03772     try{
03773         System.out.println("El vector tiene : " + lineas.size() + " lineas")
03774 ;
03775         while(!lineas.isEmpty()){
03776
03777
03778             System.out.println("Archivo No. " + ind + " - " + lineas.firstElement().toString());
03779
03780             //Aqui se obtendrán los datos y se realizarán los nuevos archivos
03781
03782             tmpLineas = lineas.firstElement().toString();
03783
03784             //Creamos una condicional para saber si se trata de un directorio
03785
03786             //y en caso de que si, eliminar las etiquetas
03787
03788             tmpLineas = tmpLineas.substring(1, tmpLineas.length()-1);
03789
03790             if(!tmpLineas.endsWith("/"))
03791                 imprimir.addElement(tmpLineas);
03792
03793             lineas.removeElementAt(d);
03794             ind++;
```

```
03795     }
03796
03797     //Todo salio bien
03798
03799     return imprimir;
03800
03801     }catch(Exception g){System.out.println(g);}
03802
03803
03804
03805     throw new Exception("No se realizó la selección");
03806 }
03807
03808
03809 private void realizarTransferencia2() throws Exception
03810 {
03811     String tmpEnvia = "";
03812     String rutaServicioWeb = "";
03813     String contenido = "";
03814     byte t[] = new byte[1024];
03815     int ind = 0;
03816
03817
03818
03819     if(selSincGeneral.equals("Sincronización con reemplazo y preguntar en cas
o de reemplazo")){
03820         try{
03821
03822             while(!salidaSeleccion1.isEmpty()){
03823
03824                 espera = true;
03825                 reemplaza = false;
03826
03827                 tmpEnvia = salidaSeleccion1.firstElement().toString();
03828
03829                 //copiamos el nombre a la variable nombreAlerta para
03830                 //enviarla a la Alerta de reemplazo de archivos
03831                 nombreAlerta = tmpEnvia;
03832
03833                 //Checamos si existe ek archivo creando una conexion
03834                 //previamente antes de ir por el contenido
03835
03836                 FileConnection fcExiste = (FileConnection)
03837                 Connector.open("file:///root1/" + tmpEnvia);
03838                 if(fcExiste.exists()) {
03839                     switchDisplayable(null, getAlertaArchivos());
03840                     while(espera)
03841                     {
03842                         textBoxAlerta.isShown();
03843                     }
03844                 }
03845                 else
03846                     reemplaza = true;
03847                 if(reemplaza){
03848                     //creamos la ruta con la información que se va a envi
ar
03849                     rutaServicioWeb = urlDefinida + "/XMLTest/resources/a
```

```

    rchivoport/descarga/?nombreakchivo=" +
03850      URLEncode(tmpEnvia) + "&usuario=" + usuarioAc
    tual +
03851      "&password=" + passwordActual;
03852
03853      //recibimos la información del WS una vez que comparó
    la bitacora local enviada
03854      //con la bitocora del servidor
03855      contenido = Parser.ConectorWeb.sendHttpGet(rutaServic
    ioWeb);
03856
03857      System.out.println("Lo que recibí de archivo:" + con
    tenido + ":");
03858
03859
03860      //paso la información recibida a un byte[]
03861
03862      t = contenido.getBytes();
03863
03864      //Creo el archivo que será el que me ayudará con la t
    ransferencia
03865
03866      archivo(t, tmpEnvia);
03867
03868      System.out.println("Se ha creado el archivo : " + tmp
    Envia);
03869    }
03870
03871      //eliminamos el que ya creamos
03872      salidaSeleccion1.removeElementAt(ind);
03873    }
03874
03875      //Todo salió bien
03876      return;
03877    }catch(Exception d){System.out.println(d);}
03878  }
03879  }
03880  else{
03881    try{
03882
03883      while(!salidaSeleccion1.isEmpty()){
03884
03885        tmpEnvia = salidaSeleccion1.firstElement().toString();
03886
03887        //creamos la ruta con la información que se va a enviar
03888        rutaServicioWeb = urlDefinida + "/XMLTest/resources/archi
    voport/descarga/?nombreakchivo=" +
03889        URLEncode(tmpEnvia) + "&usuario=" + usuarioActual
    +
03890        "&password=" + passwordActual;
03891
03892        System.out.println("ruta : " + rutaServicioWeb + ":");
03893
03894
03895        //recibimos la información del WS una vez que comparó la
    bitacora local enviada
03896        //con la bitocora del servidor

```

```
03897         contenido = Parser.ConectorWeb.sendHttpGet(rutaServicioWe
03898     b);
03898
03899         System.out.println("Lo que recibió de archivo:" + conteni
03900     do + ":");
03900
03901
03902         //paso la información recibida a un byte[]
03903
03904         t = contenido.getBytes();
03905
03906         //Creo el archivo que será el que me ayudará con la trans
03907     ferencia
03907
03908         archivo(t, tmpEnvia);
03909
03910         System.out.println("Se ha creado el archivo : " + tmpEnvi
03911     a);
03911
03912         //eliminamos el que ya creamos
03913         salidaSeleccion1.removeElementAt(ind);
03914     }
03915
03916     //Todo salio bien
03917     return;
03918 } catch (Exception d) {System.out.println(d);}
03919 }
03920
03921
03922     throw new Exception("No se realizó la selección");
03923 }
03924
03925 private void realizarTransferencia3() throws Exception
03926 {
03927     String tmpEnvia = "";
03928     String rutaServicioWeb = "";
03929     String contenido = "";
03930     String rutaServicioWebExiste = "";
03931     String contenidoExiste = "";
03932     int ind = 0;
03933     int tam = 0;
03934     int oks = 0;
03935
03936     tam = salidaSeleccion2.size();
03937
03938     byte[] c;
03939     String enviaInformacion = "";
03940
03941     if(selSincGeneral.equals("Sincronización con reemplazo y preguntar en cas
03942     o de reemplazo")){
03943         try{
03944             while(!salidaSeleccion2.isEmpty()){
03945
03946                 espera = true;
03947                 reemplaza = false;
03948
```



```

03949         tmpEnvia = salidaSeleccion2.firstElement().toString();
03950
03951         //Leemos el archivo, para enviar en una variable el conte
nido del archivo
03952
03953
03954         //copiamos el nombre a la variable nombreAlerta para
03955         //enviarla a la Alerta de reemplazo de archivos
03956         nombreAlerta = tmpEnvia;
03957
03958         //Checamos si existe ek archivo creando una conexion
03959         //previamente antes de ir por el contenido
03960
03961         FileConnection fcExiste = (FileConnection)
03962         Connector.open("file:///root1/" + tmpEnvia);
03963
03964         //crea la ruta web del servicio que verifica si existe
03965         //en el servidor
03966
03967         rutaServicioWebExiste = urlDefinida + "/XMLTest/resources
/existeport/verificaarchivo/?nombre=" +
03968         URLEncode(tmpEnvia) + "&usuario=" + usuarioActual
+ "&password=" + passwordActual;
03969
03970         //guarda en contenido. true si ya existe en el servidor
03971         //false si no existe.
03972         contenidoExiste = Parser.ConectorWeb.sendHttpGet (rutaServ
icioWebExiste);
03973
03974         if(contenidoExiste.equals("true")) {
03975             switchDisplayable(null, getAlertaArchivos());
03976             while(espera)
03977             {
03978                 textBoxAlerta.isShown();
03979             }
03980         }
03981         else
03982             reemplaza = true;
03983         if(reemplaza){
03984             c = new byte[(int)fcExiste.fileSize()];
03985             InputStream isSube = fcExiste.openInputStream();
03986
03987             isSube.read(c, 0, (int)fcExiste.fileSize());
03988             enviaInformacion = c.toString();
03989
03990             System.out.println("Lo que tiene el archivo " + tmp
Envia.toString() +
03991             " es : " + enviaInformacion.toString() + ":
");
03992
03993
03994         //creamos la ruta con la información que se va a envi
ar
03995         rutaServicioWeb = urlDefinida + "/XMLTest/resources/r
ecibeport/creararchivo/?nombre=" +
03996         URLEncode(tmpEnvia) + "&contenido=" + c +
03997         "&usuario=" + usuarioActual + "&password=" +

```



```

        passwordActual;
03998
03999 //recibimos la información del WS una vez que subimos
        la información del archivo
04000 //con la bitocora del servidor
04001 contenido = Parser.ConectorWeb.sendHttpGet(rutaServic
ioWeb);
04002
04003         if(contenido.equals("true"))
04004             oks++;
04005
04006         System.out.println("Lo que recibió de archivo:" + con
tenido + ":");
04007         isSube.close();
04008         fcExiste.close();
04009     }
04010
04011     //eliminamos el que ya creamos
04012     salidaSeleccion2.removeElementAt(ind);
04013
04014 }
04015
04016     System.out.println("tam " + tam + " " + " oks " + oks);
04017
04018     //Todo salio bien
04019     //if(tam == oks)
04020     return;
04021
04022     }catch(Exception d){System.out.println(d);}
04023 }
04024 else{
04025     try{
04026         while(!salidaSeleccion2.isEmpty()){
04027
04028             tmpEnvia = salidaSeleccion2.firstElement().toString();
04029
04030             //Leemos el archivo, para enviar en una variable el conte
nido del archivo
04031
04032             FileConnection fcSube = (FileConnection)
Connector.open("file:///root1/" + tmpEnvia);
04033             if(!fcSube.exists()) {
04034                 throw new IOException("File does not exist");
04035             }
04036
04037             c = new byte[(int)fcSube.fileSize()];
04038             InputStream isSube = fcSube.openInputStream();
04039
04040             isSube.read(c, 0, (int)fcSube.fileSize());
04041             enviaInformacion = c.toString();
04042             //enviaInformacion = new String(c,0,size);
04043
04044             System.out.println("Lo que tiene el archivo " + tmpEnvi
a.toString() +

```

```

04049         " es : " + enviaInformacion.toString() + ":";
04050
04051         //creamos la ruta con la información que se va a enviar
04052         rutaServicioWeb = urlDefinida + "/XMLTest/resources/recib
eport/creararchivo/?nombre=" +
04053         URLEncoder.encode(tmpEnvia) + "&contenido=" + c +
04054         "&usuario=" + usuarioActual + "&password=" + pass
wordActual;
04055
04056         //recibimos la información del WS una vez que subimos la
información del archivo
04057         //con la bitocora del servidor
04058         contenido = Parser.ConectorWeb.sendHttpGet(rutaServicioWe
b);
04059
04060         if(contenido.equals("true"))
04061             oks++;
04062
04063         System.out.println("Lo que recibió de archivo:" + conteni
do + ":");
04064
04065         //eliminamos el que ya creamos
04066         salidaSeleccion2.removeElementAt(ind);
04067         isSube.close();
04068         fcSube.close();
04069
04070
04071     }
04072
04073
04074     System.out.println("tam " + tam + " " + " oks " + oks);
04075
04076     //Todo salio bien
04077     if(tam == oks)
04078         return;
04079
04080     }catch(Exception d){System.out.println(d);}
04081 }
04082
04083     throw new Exception("No se realizó la selección");
04084 }
04085
04086 String readLine(Reader r) throws Exception
04087 {
04088     int c = r.read();
04089     if( c == -1 )
04090         return null;
04091
04092     String s = "";
04093     while(c!=-1 && c!= '\n' && c!= '\r')
04094     {
04095         s += (char) c;
04096         c = r.read();
04097     }
04098     return s;
04099 }
04100

```

```
04101
04102     private boolean creaBitacora( byte[] data)
04103     {
04104         String path = "bitacora.xml";
04105
04106         if( File.writeFile(path, data))
04107         {
04108             return true;
04109         }else return false;
04110     }
04111
04112     private boolean archivo( byte[] data)
04113     {
04114         String path = "archivos.txt";
04115
04116         if( File.writeFile(path, data))
04117         {
04118             return true;
04119         }else return false;
04120     }
04121
04122     private boolean archivo( byte[] data, String nombreArchivo)
04123     {
04124         String path = nombreArchivo;
04125
04126         if( File.writeFile(path, data))
04127         {
04128             return true;
04129         }else return false;
04130     }
04131
04132     private void login() throws Exception {
04133         //#ifdef LoginScreenServletExample
04134         //#          //URL
04135         //#          String url = "http://200.76.55.140:8080/LoginScreenExample/" + "
04136         //#          ?username=" + getLoginScreen().getUsername() + "&password=" + getLoginScreen().ge
04137         //#          tPassword();
04138         //#
04139         //#endif
04140         //#ifdef LoginScreenServletExample
04141         //#          //Connect to the server
04142         //#          HttpConnection hc = (HttpConnection) Connector.open(url);
04143         //#          //Authentication
04144         //#          if (hc.getResponseCode() == HttpConnection.HTTP_OK) {
04145         //#              hc.close();
04146         //#              return;
04147         //#          }
04148         //#          //Closing time...
04149         //#          hc.close();
04150         //#          //Take action based on login value
04151         //#endif
04152         //#Conectarnos a la base de datos en mysql
04153
04154         Sha1 sha = new Sha1();
04155
04156         String stringA = "";
```

```

04156         String stringB = "";
04157         String url = "propiedades.xml";
04158         String geturl = "";
04159         String rutaServicioWeb = "";
04160         String contenido = "";
04161
04162
04163         Document docurl = Parser.RestParser.getDocumentLocal("file:///root1/" +
url);
04164         urlDefinida = Parser.RestParser.beginParsePropiedad(docurl).getUrlServid
or().toString();
04165         //urlDefinida = "http://localhost:8084";
04166
04167
04168         System.out.println("url: " + urlDefinida);
04169
04170
04171         //ultima
04172         //Document doc = Parser.RestParser.getDocument(urlDefinida + "/XMLTest/u
suarios.xml");
04173         //stringA = Parser.RestParser.beginParseUser(doc).getUsername().toStrin
g
();
04174         //stringB = Parser.RestParser.beginParseUser(doc).getPassword().toStrin
g
();
04175
04176
04177         //Document doc = Parser.RestParser.getDocument("http://localhost:8084/X
MLTest/usuarios.xml");
04178
04179         //Document doc = Parser.RestParser.getDocument("http://localhost:8084/XM
LTest/usuarios.xml");
04180         //Document doc = Parser.RestParser.getDocument("http://10.0.1.91:8084/XM
LTest/usuarios.xml"); //casa
04181         //Document doc = Parser.RestParser.getDocument("http://10.1.1.10:8080/NF
FileServer/resources/users/" + getLoginScreen().getUsername() + "/" );
04182
04183         //Guardo el nombre del usuario actual
04184         usuarioActual = getLoginScreen().getUsername().toString();
04185
04186         //Guardo el password del usuario actual
04187         passwordActual = sha.getHash(getLoginScreen().getPassword()).substring(0
, 40);
04188
04189         //Parser.RestParser.setDocumentLocal("http://localhost:8085");
04190
04191         //User usuariologin = new User();
04192
04193         //ifndef LoginScreenServletExample
04194         //if the username/password starts with "test" then it's OK
04195
04196         //if(getLoginScreen().getUsername().equals("rodrigo") && getLoginScreen()
.getPassword().equals("rod")) {
04197         //if(getLoginScreen().getUsername().equals(stringA) && sha.getHash(getLo
ginScreen().getPassword()).equals(stringB)) {
04198
04199         rutaServicioWeb = urlDefinida +
04200         "/XMLTest/resources/userport/valida/?user=" +

```

```
04201         usuarioActual +
04202         "&pass=" +
04203         passwordActual;
04204
04205
04206         //http://localhost:8084/XMLTest/resources/userport/valida/?user=rodrigo&
pass=82eebalf2783a67ccb954elfe51e2ee863084784
04207
04208         contenido = Parser.ConectorWeb.sendHttpGet(rutaServicioWeb);
04209
04210         if(contenido.equals("true")) {
04211             return;
04212         }
04213         //endif
04214         //throw exception
04215         throw new Exception("No se pudo acceder");
04216     }
04217
04218
04219     private void creacionBitacora() throws Exception {
04220
04221         //if ( creacionBitacoraLocal() == true && creacionBitacoraRemota() ==
true) {
04222         if ( creacionBitacoraLocal() == true) {
04223             return;
04224         }
04225
04226         throw new Exception("No se creo la bitacora remota");
04227     }
04228
04229
04230
04231
04232     private boolean creacionBitacoraLocal() {
04233
04234         byte[] a = null;
04235
04236         xmlarchivolocal = "";
04237         try{
04238             obtenerRaiz();
04239             a = xmlarchivolocal.getBytes();
04240             creaBitacora(a);
04241             return true;
04242         }catch(Exception e){
04243             System.out.println(e);
04244         }
04245
04246         return false;
04247     }
04248
04249     private boolean creacionBitacoraRemota() {
04250
04251
04252         //Una idea seria acceder directamente a un archivo ya creado y descargarl
o
04253         //como bitacora remota siendo exito la descarga
04254         System.out.println("Entró a la pre-descarga");
```



```

04255     try{
04256         if(descargaBitacoraRemota())
04257             return true;
04258     }catch(Exception e){
04259         System.out.println(e);
04260     }
04261
04262     return false;
04263 }
04264
04265 private void comparaBitacorasDown(String pathArchivo) throws Exception
04266 {
04267     //Abrir ambos bitacoras
04268     //realizar la comparación de bitacoras
04269     //crear el archivo de archivos a transferir
04270     //cerrar archivos
04271
04272     //abrir el archivo de bitacora local para enviar por WS
04273
04274     byte c[] = new byte[1024];
04275     byte d[] = new byte[1024];
04276     String salida = null;
04277     String enviaInformacion = "";
04278     String recibeInformacion = "";
04279     String rutaServicioWeb = "";
04280
04281
04282     try {
04283         FileConnection fc = (FileConnection)
04284             Connector.open("file:///root1/" + pathArchivo);
04285         if(!fc.exists()) {
04286             throw new IOException("File does not exist");
04287         }
04288         InputStream is = fc.openInputStream();
04289         byte b[] = new byte[1024];
04290         int length = is.read(b, 0, 1024);
04291
04292         enviaInformacion = new String(b, 0 , length);
04293
04294         System.out.println("Lo que leyó de la bitacora local:\n" + en
04295 viaInformacion.toString() + ":");
04296
04297         if(selSincGeneral.equals("Sincronización sin reemplazo")){
04298             //creamos la ruta con la información que se va a enviar
04299
04300             //si solo queremos los diferentes
04301             rutaServicioWeb = urlDefinida + "/XMLTest/resources/bit
04302 acorasport/comparabitacoras/?arbolArchivos=" +
04303 enviaInformacion + "&usuario=" + usuarioActual
04304 +
04305             "&password=" + passwordActual;
04306         }
04307         else if(selSincGeneral.equals("Sincronización con reemplazo
04308 ")){
04309             //creamos la ruta con la información que se va a enviar

```



```

04306             //si queremos reemplazo, es decir, va por todos
04307             rutaServicioWeb = urlDefinida + "/XMLTest/resources/arc
hivosreemplazoport/lista/?usuario=" +
04308                 usuarioActual + "&password=" + passwordActual;
04309         }
04310         else if (selSincGeneral.equals("Sincronización con reemplazo
y preguntar en caso de reemplazo")){
04311             //creamos la ruta con la información que se va a enviar

04312             //si queremos reemplazo, es decir, va por todos
04313             rutaServicioWeb = urlDefinida + "/XMLTest/resources/arc
hivosreemplazoport/lista/?usuario=" +
04314                 usuarioActual + "&password=" + passwordActual;
04315         }
04316
04317         //recibimos la información del WS una vez que comparó la bi
tadora local enviada
04318         //con la bitadora del servidor
04319         recibeInformacion = Parser.ConectorWeb.sendHttpGet(rutaServ
icioWeb);
04320
04321         System.out.println("Lo que leyó de la bitadora local:" + re
cibeInformacion + ":");
04322
04323         //paso la información recibida a un byte[]
04324
04325         c = recibeInformacion.getBytes();
04326
04327         //Creo el archivo que será el que me ayudará con la transfe
rencia
04328
04329         archivo(c, "transfer.xml");
04330
04331         is.close();
04332         fc.close();
04333
04334         //Todo bien
04335         return;
04336
04337     } catch (IOException m) {System.out.println(m);}
04338
04339     throw new Exception("No se creo la bitadora remota");
04340 }
04341
04342 private void comparaBitadorasUp(String pathArchivo1) throws Exception
04343 {
04344     //Abrir ambos bitadoras
04345     //realizar la comparación de bitadoras
04346     //crear el archivo de archivos a transferir
04347     //cerrar archivos
04348
04349     //abrir el archivo de bitadora local para enviar por WS
04350
04351     byte c[] = new byte[1024];
04352     byte d[] = new byte[1024];
04353     String salida = null;
04354     String enviaInformacion="";

```

```
04355         String recibeInformacion = "";
04356         String rutaServicioWeb = "";
04357
04358
04359
04360         try {
04361
04362             FileConnection fc = (FileConnection)
04363             Connector.open("file:///root1/" + pathArchivo1);
04364             if(!fc.exists()) {
04365                 throw new IOException("File does not exist");
04366             }
04367             InputStream is = fc.openInputStream();
04368             byte b[] = new byte[1024];
04369             int length = is.read(b, 0, 1024);
04370
04371             enviaInformacion = new String(b, 0, length);
04372
04373             System.out.println("Lo que leyó de la bitacora local:\n" + en
viaInformacion.toString() + ":");
04374
04375
04376
04377
04378             if(selSincGeneral.equals("Sincronización sin reemplazo")){
04379                 //creamos la ruta con la información que se va a enviar
04380
04381                 rutaServicioWeb = urlDefinida + "/XMLTest/resources/com
parabitacoraupport/comparabitacoras/?arbolArchivos=" +
04382                 enviaInformacion + "&usuario=" + usuarioActual
04383                 +
04384                 "&password=" + passwordActual;
04385             }
04386             else if (selSincGeneral.equals("Sincronización con reemplaz
o")){
04387                 //creamos la ruta con la información que se va a enviar
04388
04389                 rutaServicioWeb = urlDefinida + "/XMLTest/resources/esp
ejoport/igual/?arbolarchivos=" +
04390                 enviaInformacion;
04391
04392             }
04393             else if (selSincGeneral.equals("Sincronización con reemplazo
y preguntar en caso de reemplazo")){
04394
04395                 //creamos la ruta con la información que se va a enviar
04396
04397                 rutaServicioWeb = urlDefinida + "/XMLTest/resources/esp
ejoport/igual/?arbolarchivos=" +
04398                 enviaInformacion;
04399
04400             }
04401             System.out.println("La ruta es la siguiente: " + rutaServic
```

```

        ioWeb + ":"");
04402                //recibimos la información del WS una vez que comparó la bi
        tacora local enviada
04403                //con la bitocora del servidor
04404                recibeInformacion = Parser.ConectorWeb.sendHttpGet (ruta
        ServicioWeb);
04405
04406                System.out.println("Lo que leyó del WS1:" + recibeInfor
        macion + ":"");
04407
04408                //paso la información recibida a un byte[]
04409
04410                d = recibeInformacion.getBytes();
04411
04412                //Creo el archivo que será el que me ayudará con la tran
        sferencia
04413
04414                archivo(d, "transfer.xml");
04415
04416
04417                is.close();
04418                fc.close();
04419
04420                //Todo bien
04421                return;
04422
04423                }catch (IOException m){System.out.println(m);}
04424
04425                throw new Exception("No se creo la bitacora remota");
04426        }
04427
04428
04429
04430        private boolean descargaBitacoraRemota()
04431        {
04432
04433                //realizar la descarga de la bitacora
04434                //posible uso de crearArchivo si solo se obtiene el contenido
04435
04436                System.out.println("Si entró a descargarla ...");
04437
04438
04439                String id = "";
04440                byte[] contenido = null;
04441
04442                Document descarga = Parser.RestParser.getDocument (urlDefinida + "/XMLTest
        /archivo.xml");
04443
04444                id = Parser.RestParser.beginParseFileContent (descarga).getId().toString()
        ;
04445                contenido = Parser.RestParser.beginParseFileContent (descarga).getContent (
        );
04446
04447
04448                System.out.println("Pasó la lectura ...");
04449
04450                if (archivo (contenido, "bitacoraRem.xml"))

```

```

04451     {
04452         System.out.println("Ya esta la bitacora remota");
04453         return true;
04454     }
04455
04456     return false;
04457 }
04458
04459 private void muestraPropiedades(String path, String filename)
04460 {
04461     try
04462     {
04463         FileConnection fc = (FileConnection) Connector.open("file:///root1/"
+ filename);
04464
04465         double tamaño = (double) (fc.fileSize());
04466         String tam = "";
04467         if (tamaño > 1024)
04468         {
04469             tamaño /= 1024;
04470             tam = " kb";
04471         }
04472         else
04473         {
04474             tam = " bytes";
04475         }
04476
04477         Alert props = new Alert(filename,
04478             "Ultima Modificacion " + new Date(fc.lastModified()) + "\n" +
04479             "Permiso de Escritura: " + (fc.canWrite() ? "Si" : "No") + "\n" +
n" +
04480             "Permiso de Lectura: " + (fc.canRead() ? "Si" : "No") + "\n"
+
04481             "Tamaño del archivo: " + tamaño + tam ,
04482             null, AlertType.WARNING);
04483         props.setTimeout(Alert.FOREVER);
04484         Display.getDisplay(this).setCurrent(props);
04485         fc.close();
04486     }catch(IOException ioe){}
04487 }
04488
04489 private String muestraArchivoTexto(String pathfileName) {
04490
04491     System.out.println(pathfileName);
04492     if(!pathfileName.endsWith(".txt"))
04493         return "El Visor de Archivos solo soporta archivos de texto (.txt)";
04494
04495     byte b[] = new byte[1024];
04496     String salida = null;
04497     try {
04498         FileConnection fc = (FileConnection)
04499             Connector.open("file:///root1/" + pathfileName);
04500         if(!fc.exists()) {
04501             throw new IOException("File does not exist");
04502         }
04503         InputStream is = fc.openInputStream();

```

```
04504         is.read(b, 0, 1024);
04505         //int length = is.read(b, 0, 1024);
04506         System.out.println("texto:" + b.toString() + ":");
04507         salida = new String(b);
04508         System.out.println("texto:" + salida + ":");
04509         // Alert texto = new Alert(pathfileName,new String(b, 0, length), null,
AlertType.INFO);
04510         // texto.setTimeout(Alert.FOREVER);
04511         // Display.getDisplay(this).setCurrent(texto);
04512         is.close();
04513         fc.close();
04514         } catch (Exception e) {
04515             System.out.println(e);
04516         }
04517         return salida;
04518     }
04519
04520
04521
04522     private void muestraArchivoAlerta(String pathfileName) {
04523
04524         if(!pathfileName.endsWith(".txt"))
04525         {
04526             Alert filetext = new Alert(pathfileName,
04527                 "El Visor de Archivos solo soporta archivos de texto (.txt)",
04528
04529                 null, AlertType.WARNING);
04529             filetext.setTimeout(Alert.FOREVER);
04530             Display.getDisplay(this).setCurrent(filetext);
04531         }
04532         else
04533         {
04534
04535             byte b[] = new byte[1024];
04536             String salida = null;
04537             try {
04538                 FileConnection fc = (FileConnection)
04539                     Connector.open("file:///root1/" + pathfileName);
04540                 if(!fc.exists()) {
04541                     throw new IOException("File does not exist");
04542                 }
04543                 InputStream is = fc.openInputStream();
04544                 is.read(b, 0, 1024);
04545                 //int length = is.read(b, 0, 1024);
04546                 System.out.println("texto:" + b.toString() + ":");
04547                 salida = new String(b);
04548                 System.out.println("texto:" + salida + ":");
04549                 Alert filetext = new Alert(pathfileName,
04550                     salida,
04551                     null, AlertType.WARNING);
04552                 filetext.setTimeout(Alert.FOREVER);
04553                 Display.getDisplay(this).setCurrent(filetext);
04554                 is.close();
04555                 fc.close();
04556             } catch (Exception e) {
04557                 System.out.println(e);
04558             }
04559         }
04560     }
04561 }
```



```

04559     }
04560
04561   }
04562
04563   private void obtenerRaiz() {
04564       Enumeration drives = FileSystemRegistry.listRoots();
04565       //System.out.println("Las raices son las siguientes: ");
04566       while(drives.hasMoreElements()) {
04567           String root = (String) drives.nextElement();
04568           System.out.println("\n" + root);
04569           //xmlarchivolocal += "<" + root + ">\n";
04570           xmlarchivolocal += "<" + root + ">";
04571           obtenerListadoArchivos(root);
04572       }
04573   }
04574
04575
04576   private void obtenerListadoArchivos(String directorio) {
04577       // System.out.println("Estoy aqui");
04578       try {
04579
04580           FileConnection fc = (FileConnection)Connector.open("file:/// " + director
04581 io);
04582           System.out.println("Lista de archivos:");
04583           Enumeration filelist = fc.list("/*", true);
04584           while(filelist.hasMoreElements()) {
04585               String fileName = (String) filelist.nextElement();
04586               fc = (FileConnection)Connector.open("file:/// " + directorio + fileNam
04587 e);
04588               if(fc.isDirectory()) {
04589                   System.out.println("\tDirectorio: " + fileName);
04590                   xmlarchivolocal += "<" + fileName + ">";
04591                   obtenerListadoArchivos(directorio + fileName);
04592               } else {
04593                   if(!fileName.equals("bitacora.xml") ){
04594                       if(!fileName.equals("propiedades.xml") ){
04595                           if(!fileName.equals("transfer.xml") ){
04596                               if(!fileName.equals("desktop.ini") ){
04597                                   System.out.println("\tArchivo: " + fileName);
04598                                   xmlarchivolocal += "<" + fileName + ">";
04599                               }
04600                           }
04601                       }
04602                   }
04603               }
04604               fc.close();
04605               System.out.println(xmlarchivolocal);
04606           } catch (IOException ioe) {
04607               System.out.println(ioe);
04608           }
04609       }
04610
04611   private String URLEncode(String s)
04612   {
04613       StringBuffer sbuf = new StringBuffer();
04614       int ch;

```



```
04614     for (int i = 0; i < s.length(); i++)
04615     {
04616         ch = s.charAt(i);
04617         switch(ch)
04618         {
04619             case '+': { sbuf.append("+"); break;}
04620             case '!': { sbuf.append("%21"); break;}
04621             case '*': { sbuf.append("%2A"); break;}
04622             case '\\': { sbuf.append("%27"); break;}
04623             case '{': { sbuf.append("%28"); break;}
04624             case ')': { sbuf.append("%29"); break;}
04625             case ';': { sbuf.append("%3B"); break;}
04626             case ':': { sbuf.append("%3A"); break;}
04627             case '@': { sbuf.append("%40"); break;}
04628             case '&': { sbuf.append("%26"); break;}
04629             case '=': { sbuf.append("%3D"); break;}
04630             case '+': { sbuf.append("%2B"); break;}
04631             case '$': { sbuf.append("%24"); break;}
04632             case ',': { sbuf.append("%2C"); break;}
04633             case '/': { sbuf.append("%2F"); break;}
04634             case '?': { sbuf.append("%3F"); break;}
04635             case '%': { sbuf.append("%25"); break;}
04636             case '#': { sbuf.append("%23"); break;}
04637             case '[': { sbuf.append("%5B"); break;}
04638             case ']': { sbuf.append("%5D"); break;}
04639             default: sbuf.append((char)ch);
04640         }
04641     }
04642     return sbuf.toString();
04643 }
04644
04645
04646
04647
04648 }
04649
```

6.13 src/Parser/ConectorWeb.java File Reference

Classes

- class [Parser.ConectorWeb](#)

Packages

- package [Parser](#)

6.14 src/Parser/ConectorWeb.java

```
00001 /*
00002  * To change this template, choose Tools | Templates
00003  * and open the template in the editor.
00004  */
00005
00006 package Parser;
00007
00008 import java.io.DataInputStream;
00009 import java.io.IOException;
00010 import javax.microedition.io.Connector;
00011 import javax.microedition.io.HttpConnection;
00012
00013
00014 public class ConectorWeb {
00015
00016     public static String sendHttpGet( String url ) throws IOException {
00017
00018         HttpConnection hcon = null;
00019         DataInputStream dis = null;
00020         String message = "";
00021
00022         try {
00023             int ch;
00024
00025             hcon = ( HttpConnection ) Connector.open( url );
00026             dis = new DataInputStream( hcon.openInputStream() );
00027
00028             while ( ( ch = dis.read() ) != -1 ) {
00029                 message += ( char )ch;
00030             }finally {
00031                 if ( hcon != null ) hcon.close();
00032                 if ( dis != null ) dis.close();
00033             }
00034
00035             return message.toString();
00036         }
00037     }
00038 }
00039
00040
00041
00042
00043
00044
00045 }
```

6.15 src/Parser/RestParser.java File Reference

Classes

- class [Parser.RestParser](#)

Packages

- package [Parser](#)

6.16 src/Parser/RestParser.java

```
00001 /*
00002  * To change this template, choose Tools | Templates
00003  * and open the template in the editor.
00004  */
00005
00006 package Parser;
00007
00008
00009 import Core.File;
00010 import Core.FileContent;
00011 import Core.License;
00012 import Core.Propiedades;
00013 import Core.User;
00014 import java.io.IOException;
00015 import java.io.InputStream;
00016 import java.io.InputStreamReader;
00017 import java.io.OutputStream;
00018 import java.io.PrintStream;
00019 import java.util.Vector;
00020 import javax.microedition.io.*;
00021 import javax.microedition.io.HttpConnection;
00022 import javax.microedition.io.file.FileConnection;
00023
00024 import javax.microedition.lcdui.DateField;
00025 import org.kxml.parser.*;
00026 import org.kxml.kdom.*;
00027 import org.kxml.*;
00028
00029
00034 public class RestParser {
00035
00043     public static User beginParseUser(Element root)
00044     {
00045         int child_count = root.getChildCount();
00046         User user = new User();
00047
00048         for (int i = 0; i < child_count ; i++ ) {
00049             if (root.getType(i) != Xml.ELEMENT) {
00050                 continue;
00051             }
00052
00053             Element kid = root.getElement(i);
00054
00055             // it's not an address? continue
00056             if (kid.getName().equals("id")) {
00057
00058                 user.setId(Integer.valueOf(kid.getText()));
00059             }
00060             else if(kid.getName().equals("password")) {
00061
00062                 user.setPassword(kid.getText());
00063             }
00064             else if(kid.getName().equals("username")) {
00065
00066                 user.setUsername(kid.getText());
```

```

00067         }
00068         else if(kid.getName().equals("licenseCollection")) {
00069             int kidslength = kid.getChildCount();
00070             for(int j = 0; j < kidslength; j++)
00071             [
00072                 user.getLicenseCollection().addElement(((Element)kid.g
etChild(j)).getAttribute("uri").getValue());
00073             ]
00074         }
00075     }
00076     }
00077     kid = null;
00078 }
00079 return user;
00080 }
00081
00082 public static User beginParseUser(Document doc) {
00083     Element root = doc.getRootElement();
00084     return beginParseUser(root);
00085 }
00086
00087 public static Vector beginParseUsers(Document doc)
00088 {
00089     Vector users = new Vector();
00090
00091     Element root = doc.getRootElement();
00092     int child_count = root.getChildCount();
00093
00094     for (int i = 0; i < child_count ; i++ ) {
00095         if (:root.getType(i) != Xml.ELEMENT) {
00096             continue;
00097         }
00098
00099         Element kid = root.getElement(i);
00100
00101         // it's not an address? continue
00102         if (kid.getName().equals("user")) {
00103
00104             User us = beginParseUser(kid);
00105             if(us != null)
00106                 users.addElement(us.toString());
00107
00108             kid = null;
00109         }
00110     }
00111     return users;
00112 }
00113
00114 public static License beginParseLicense(Element root)
00115 {
00116
00117     int child_count = root.getChildCount();
00118     License license = new License();

```



```
00136
00137     for (int i = 0; i < child_count ; i++ ) {
00138         if (root.getType(i) != Xml.ELEMENT) {
00139             continue;
00140         }
00141
00142         Element kid = root.getElement(i);
00143
00144         // it's not an address? continue
00145         if (kid.getName().equals("id")) {
00146             license.setId(Integer.valueOf(kid.getText()));
00147         }
00148         else if(kid.getName().equals("type")) {
00149             license.setType(kid.getText());
00150         }
00151         else if(kid.getName().equals("fileid")) {
00152             license.setFileid(kid.getAttribute("uri").getValue());
00153         }
00154         else if(kid.getName().equals("username")) {
00155             license.setUserid(kid.getAttribute("uri").getValue());
00156         }
00157     }
00158     kid = null;
00159 }
00160 return license;
00161 }
00162 }
00163
00164 public static License beginParseLicense(Document doc) {
00165     Element root = doc.getRootElement();
00166     return beginParseLicense(root);
00167 }
00168
00169 public static Vector beginParseLicenses(Document doc)
00170 {
00171     Vector licenses = new Vector();
00172
00173     Element root = doc.getRootElement();
00174     int child_count = root.getChildCount();
00175
00176     for (int i = 0; i < child_count ; i++ ) {
00177         if (root.getType(i) != Xml.ELEMENT) {
00178             continue;
00179         }
00180
00181         Element kid = root.getElement(i);
00182
00183         // it's not an address? continue
00184         if (kid.getName().equals("license")) {
```

```

00206
00207         license lic = beginParseLicense(kid);
00208         if(lic != null)
00209             licenses.addElement(lic);
00210     }
00211     kid = null;
00212 }
00213     return licenses;
00214 }
00215
00223 public static File beginParseFile(Element root)
00224 {
00225     int child_count = root.getChildCount();
00226     File file = new File();
00227
00228
00229
00230
00231     for (int i = 0; i < child_count ; i++ ) {
00232         if (root.getType(i) != Xml.ELEMENT) {
00233             continue;
00234         }
00235
00236         Element kid = root.getElement(i);
00237
00238         // it's not an address? continue
00239         if (kid.getName().equals("id")) {
00240             file.setId(Integer.valueOf(kid.getText()));
00241
00242         }
00243         else if(kid.getName().equals("filename")) {
00244             file.setFilename(kid.getText());
00245
00246         }
00247         else if(kid.getName().equals("filesize")) {
00248             file.setFileSize(Integer.parseInt(kid.getText()));
00249
00250         }
00251         else if(kid.getName().equals("localpath")) {
00252             file.setLocalpath(kid.getText());
00253
00254         }
00255         else if(kid.getName().equals("creationdate")) {
00256             DateField df1 = new DateField(kid.getText(), DateField.DA
TE_TIME);
00257             file.setCreationdate(df1.getDate());
00258
00259         }
00260         else if(kid.getName().equals("modificationdate")) {
00261             DateField df2 = new DateField(kid.getText(), DateField.DA
TE_TIME);
00262             file.setModificationdate(df2.getDate());
00263         }
00264
00265
00266
00267         kid = null;

```

```
00268     }
00269     return file;
00270 }
00271
00272 public static File beginParseFile(Document doc) {
00280     Element root = doc.getRootElement();
00281     return beginParseFile(root);
00282 }
00283
00284 }
00285
00286 public static Vector beginParseFiles(Document doc)
00287 {
00288     Vector licenses = beginParseLicenses(doc);
00289     Vector files = new Vector();
00290     System.out.println(licenses.size());
00291     for(int i = 0; i < licenses.size(); i++)
00292     {
00293         try{
00294             System.out.println(licenses.elementAt(i) + "\n\n");
00295             License lic = (License)licenses.elementAt(i);
00296             Document doclic = Parser.RestParser.getDocument(lic.getFileid());
00297
00298             File file = Parser.RestParser.beginParseFile(doclic);
00299             files.addElement(file);
00300         }catch(Exception e){e.printStackTrace();}
00301     }
00302
00303     return files;
00304 }
00305
00306 public static Document getDocument(String url)
00307 {
00308     XmlParser parser = null;
00309     Document doc = new Document();
00310     String str = "";
00311
00312     try {
00313         HttpURLConnection hc = null;
00314
00315         try {
00316             hc = (HttpURLConnection)Connector.open(url);
00317             //hc.openInputStream();
00318             //parse(hc.openInputStream());
00319             parser = new XmlParser( new InputStreamReader( hc.openInputStream()
00320 ));
00321
00322             doc.parse( parser );
00323         }
00324         catch (IOException ioe) {
00325             ioe.printStackTrace();
00326
00327             //mRSSListener.exception(ioe);
00328         }
00329     }
00330     finally {
00331         try { if (hc != null) hc.close(); }
00332         catch (IOException ignored) {}
00333     }
00334 }
```

```

00338         }
00339
00340     } catch (Exception ioe) {
00341         // report error
00342         System.err.println("XML Parsing Error: " + ioe);
00343         ioe.printStackTrace();
00344
00345         parser = null;
00346         doc = null;
00347
00348         return null;
00349     }
00350
00351     return doc;
00352 }
00353
00354
00355 public static Document getDocumentLocal(String url)
00356 {
00357     XmlParser parser = null;
00358     Document doc = new Document();
00359     //se abre el archivo de transferencia
00360     FileConnection fcTrans = null;
00361
00362     try {
00363
00364         try {
00365             fcTrans = (FileConnection)Connector.open(url);
00366
00367             if(!fcTrans.exists())
00368                 System.out.println("No existe tu archivo");
00369             //hc.openInputStream();
00370             //parse(hc.openInputStream());
00371             parser = new XmlParser( new InputStreamReader( fcTrans.openInputStr
eam()));
00372             doc.parse( parser );
00373         }
00374         catch (IOException ioe) {
00375             ioe.printStackTrace();
00376         }
00377         finally {
00378             try { if (fcTrans != null) fcTrans.close(); }
00379             catch (IOException ignored) {
00380                 ignored.printStackTrace();
00381             }
00382         }
00383     }
00384
00385     } catch (Exception ioe) {
00386         // report error
00387         System.err.println("XML Parsing Error: " + ioe);
00388         ioe.printStackTrace();
00389
00390         parser = null;
00391         doc = null;
00392
00393         return null;
00394     }
00395 }
00396
00397
00398
00399
00400

```

```

00401
00402     return doc;
00403 }
00404
00412 public static void setDocumentLocal(String url)
00413 {
00414     XmlParser parser = null;
00415     Document doc = new Document();
00416     String str = "";
00417
00418     str += "<?xml version=\"1.0\" encoding=\"UTF-8\"?>\n" +
00419           "\n" +
00420           "<!--\n" +
00421           "\tDocument    : propiedades.xml\n" +
00422           "\tCreated on  : 28 de octubre de 2009, 04:48 PM\n" +
00423           "\tAuthor      : Rodrigo\n" +
00424           "\tDescription:\n" +
00425           "\tXML que sirve para hacer las pruebas.\n" +
00426           "-->\n" +
00427           "\n" +
00428           "<root>\n" +
00429           "\t<urlservidor>" +
00430           url +
00431           "</urlservidor>\n" +
00432           "</root>";
00433
00434     try {
00435
00436         OutputConnection connection = (OutputConnection)
00437         Connector.open( "file:///root1/propiedades.xml", Connector.WRITE
00438 );
00439
00440         OutputStream out = connection.openOutputStream();
00441         PrintStream output = new PrintStream( out );
00442         output.println(str);
00443         out.close();
00444         connection.close();
00445
00446     }catch (Exception ioe) {
00447         // report error
00448         System.err.println("XML Parsing Error: " + ioe);
00449         ioe.printStackTrace();
00450
00451     }
00452 }
00453
00461 public static FileContent beginParseFileContent(Element root)
00462 {
00463     int child_count = root.getChildCount();
00464     FileContent filecon = new FileContent();
00465
00466     for (int i = 0; i < child_count ; i++ ) {
00467         if (root.getType(i) != Xml.ELEMENT) {
00468             continue;

```

```
00470         }
00471         Element kid = root.getElement(i);
00472         Element kid = root.getElement(i);
00473         // it's not an address? continue
00474         if (kid.getName().equals("id")) {
00475             filecon.setId(Integer.valueOf(kid.getText()));
00476         }
00477         // it's not an address? continue
00478         else if(kid.getName().equals("fileid")) {
00479             filecon.setFileid(kid.getAttribute("uri").getValue());
00480         }
00481         // it's not an address? continue
00482         else if(kid.getName().equals("content")) {
00483             byte[] content = kid.getText().getBytes();
00484             filecon.setContent(content);
00485         }
00486         kid = null;
00487     }
00488     }
00489     }
00490     }
00491     }
00492     }
00493     }
00494     }
00495     }
00496     }
00497     }
00498     }
00499     }
00500     }
00501     }
00502     }
00503     }
00504     }
00505     }
00506     }
00507     }
00508     }
00509     }
00510     }
00511     }
00512     }
00513     }
00514     }
00515     }
00516     }
00517     }
00518     }
00519     }
00520     }
00521     }
00522     }
00523     }
00524     }
00525     }
00526     }
00527     }
00528     }
00529     }
00530     }
00531     }
00532     }
00533     }
```



```
00541     public static Propiedades beginParsePropiedad(Document doc) {
00542
00543         Element root = doc.getRootElement();
00544         return beginParsePropiedad(root);
00545     }
00546
00547 }
```

6.17 src/Parser/StringTokenizer.java File Reference

Classes

- class [Parser.StringTokenizer](#)
- class [Parser.UnsupportedOperationException](#)

Packages

- package [Parser](#)

6.18 src/Parser/StringTokenizer.java

```
00001 /*
00002  * To change this template, choose Tools | Templates
00003  * and open the template in the editor.
00004  */
00005
00006 package Parser;
00007
00008 /*
00009  * A replacement for java.util.StringTokenizer
00010  * Copyright (C) 2001 Stephen Ostermiller
00011  * http://ostermiller.org/contact.pl?regarding=Java+Utilities
00012  *
00013  * This program is free software; you can redistribute it and/or modify
00014  * it under the terms of the GNU General Public License as published by
00015  * the Free Software Foundation; either version 2 of the License, or
00016  * (at your option) any later version.
00017  *
00018  * This program is distributed in the hope that it will be useful,
00019  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00020  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00021  * GNU General Public License for more details.
00022  *
00023  * See COPYING.TXT for details.
00024  */
00025
00026
00027 public class StringTokenizer implements java.util.Enumeration{
00028
00035     protected String text;
00036
00044     protected int strLength;
00045
00051     protected String nontokenDelims;
00052
00058     protected String tokenDelims;
00059
00071     protected int position;
00072
00085     protected boolean emptyReturned;
00086
00102     protected char maxDelimChar;
00103
00112     protected boolean returnEmptyTokens;
00113
00123     protected int delimsChangedPosition;
00124
00132     protected int tokenCount;
00133
00149     public StringTokenizer(String text, String nontokenDelims, String
tokenDelims){
00150         this(text, nontokenDelims, tokenDelims, false);
00151     }
00152
00175     public StringTokenizer(String text, String nontokenDelims, String
tokenDelims, boolean returnEmptyTokens){
```

```

00176         setDelims(nontokenDelims, tokenDelims);
00177         setText(text);
00178         setReturnEmptyTokens(returnEmptyTokens);
00179     }
00180
00205     public StringTokenizer(String text, String delims, boolean delimsAreToken
s){
00206         this(text, (delimsAreTokens ? null : delims), (delimsAreTokens ?
delims : null));
00207     }
00208
00222     public StringTokenizer(String text, String nontokenDelims){
00223         this(text, nontokenDelims, null);
00224     }
00225
00238     public StringTokenizer(String text){
00239         this(text, "\t\n\r\f", null);
00240     }
00241
00255     public void setText(String text){
00256         if (text == null){
00257             throw new NullPointerException();
00258         }
00259         this.text = text;
00260         strLength = text.length();
00261         emptyReturned = false;
00262         // set the position to start evaluation to zero
00263         // unless the string has no length, in which case
00264         // the entire string has already been examined.
00265         position = (strLength > 0 ? 0 : -1);
00266         // because the text was changed since the last time the delimiter
s
00267         // were changed we need to set the delimiter changed position
00268         delimsChangedPosition = 0;
00269         // The token count changes when the text changes
00270         tokenCount = -1;
00271     }
00272
00283     private void setDelims(String nontokenDelims, String tokenDelims){
00284         this.nontokenDelims = nontokenDelims;
00285         this.tokenDelims = tokenDelims;
00286         // If we change delimiters, we do not want to start fresh,
00287         // without returning empty tokens.
00288         // the delimiter changed position can never be less than
00289         // zero, unlike position.
00290         delimsChangedPosition = (position != -1 ? position : strLength);
00291         // set the max delimiter
00292         maxDelimChar = 0;
00293         for (int i=0; nontokenDelims != null && i < nontokenDelims.length
()); i++){
00294             if (maxDelimChar < nontokenDelims.charAt(i)){
00295                 maxDelimChar = nontokenDelims.charAt(i);
00296             }
00297         }
00298         for (int i=0; tokenDelims != null && i < tokenDelims.length(); i+
+){
00299             if (maxDelimChar < tokenDelims.charAt(i)){

```

```
00300         maxDelimChar = tokenDelims.charAt(i);
00301     }
00302 }
00303 // Changing the delimiters may change the number of tokens
00304 tokenCount = -1;
00305 }
00306
00307
00320 public boolean hasMoreTokens() {
00321
00322     // handle the easy case in which the number
00323     // of tokens has been counted.
00324     if (tokenCount == 0){
00325         return false;
00326     } else if (tokenCount > 0){
00327         return true;
00328     }
00329
00330     // copy over state variables from the class to local
00331     // variables so that the state of this object can be
00332     // restored to the state that it was in before this
00333     // method was called.
00334     int savedPosition = position;
00335     boolean savedEmptyReturned = emptyReturned;
00336
00337     int workingPosition = position;
00338     boolean workingEmptyReturned = emptyReturned;
00339     boolean onToken = advancePosition();
00340     while(position != workingPosition ||
00341           emptyReturned != workingEmptyReturned){
00342         if (onToken){
00343             // restore object state
00344             position = savedPosition;
00345             emptyReturned = savedEmptyReturned;
00346             return true;
00347         }
00348         workingPosition = position;
00349         workingEmptyReturned = emptyReturned;
00350         onToken = advancePosition();
00351     }
00352
00353     // restore object state
00354     position = savedPosition;
00355     emptyReturned = savedEmptyReturned;
00356     return false;
00357 }
00358
00369 public String nextToken(){
00370     int workingPosition = position;
00371     boolean workingEmptyReturned = emptyReturned;
00372     boolean onToken = advancePosition();
00373     while(position != workingPosition ||
00374           emptyReturned != workingEmptyReturned){
00375         if (onToken){
00376             // returning a token decreases the token count
00377             tokenCount--;
00378             return (emptyReturned ? "" : text.substring(worki
```

```

        ngPosition, (position != -1) ? position : strLength));
00379     }
00380         workingPosition = position;
00381         workingEmptyReturned = emptyReturned;
00382         onToken = advancePosition();
00383     }
00384     throw new java.util.NoSuchElementException();
00385 }
00386
00400 public boolean skipDelimiters(){
00401     int workingPosition = position;
00402     boolean workingEmptyReturned = emptyReturned;
00403     boolean onToken = advancePosition();
00404
00405     // skipping delimiters may cause the number of tokens to change
00406     tokenCount = -1;
00407
00408     while(position != workingPosition ||
00409         emptyReturned != workingEmptyReturned){
00410         if (onToken){
00411             // restore the state to just as it was before we
found
00412                 // this token and return
00413                 position = workingPosition;
00414                 emptyReturned = workingEmptyReturned;
00415                 return true;
00416             }
00417             workingPosition = position;
00418             workingEmptyReturned = emptyReturned;
00419             onToken = advancePosition();
00420         }
00421
00422         // the end of the string was reached
00423         // without finding any tokens
00424         return false;
00425     }
00426
00438 public int countTokens(){
00439
00440     // return the cached token count if a cache
00441     // is available.
00442     if (this.tokenCount >=0){
00443         return this.tokenCount;
00444     }
00445
00446     int tokenCount = 0;
00447
00448     // copy over state variables from the class to local
00449     // variables so that the state of this object can be
00450     // restored to the state that it was in before this
00451     // method was called.
00452     int savedPosition = position;
00453     boolean savedEmptyReturned = emptyReturned;
00454
00455     int workingPosition = position;
00456     boolean workingEmptyReturned = emptyReturned;
00457     boolean onToken = advancePosition();

```



```
00458         while(position != workingPosition ||
00459                emptyReturned != workingEmptyReturned){
00460             if (onToken){
00461                 tokenCount++;
00462             }
00463             workingPosition = position;
00464             workingEmptyReturned = emptyReturned;
00465             onToken = advancePosition();
00466         }
00467
00468         // restore object state
00469         position = savedPosition;
00470         emptyReturned = savedEmptyReturned;
00471
00472         // Save the token count in case this is called again
00473         // so we wouldn't have to do so much work.
00474         this.tokenCount = tokenCount;
00475
00476         return tokenCount;
00477     }
00478
00488     public void setDelimiters(String delims){
00487         setDelims(delims, null);
00488     }
00489
00501     public void setDelimiters(String delims, boolean delimsAreTokens){
00502         setDelims((delimsAreTokens ? null : delims), (delimsAreTokens ? d
00503 elims : null));
00504     }
00505
00513     public void setDelimiters(String nontokenDelims, String tokenDelims){
00514         setDelims(nontokenDelims, tokenDelims);
00515     }
00516
00526     public void setDelimiters(String nontokenDelims, String tokenDelims, bool
00527 ean returnEmptyTokens){
00528         setDelims(nontokenDelims, tokenDelims);
00529         setReturnEmptyTokens(returnEmptyTokens);
00530     }
00531
00545     public int countTokens(String delims){
00546         setDelims(delims, null);
00547         return countTokens();
00548     }
00549
00568     public int countTokens(String delims, boolean delimsAreTokens){
00569         setDelims((delimsAreTokens ? null : delims), (delimsAreTokens ? d
00570 elims : null));
00571         return countTokens();
00572     }
00573
00588     public int countTokens(String nontokenDelims, String tokenDelims){
00589         setDelims(nontokenDelims, tokenDelims);
00590         return countTokens();
00591     }
00592
00609     public int countTokens(String nontokenDelims, String tokenDelims, boolean
```

```

returnEmptyTokens){
00610         setDelims(nontokenDelims, tokenDelims);
00611         setReturnEmptyTokens(returnEmptyTokens);
00612         return countTokens();
00613     }
00614
00625     private boolean advancePosition(){
00626         // if we are returning empty tokens, we are just starting to tokenize
00627         // and there is a delimiter at the beginning of the string or the
00628         // string
00629         // is empty we need to indicate that there is an empty token at the
00630         // beginning.
00631         // The beginning is defined as where the delimiters were last changed.
00632         if (returnEmptyTokens && !emptyReturned &&
            (delimsChangedPosition == position ||
            position == -1 && strLength == delimsChangedPosition)){
00633             if (strLength == delimsChangedPosition){
00634                 // Case in which the string (since delim change)
00635                 // is empty, but because we are returning empty
00636                 // tokens, a single empty token should be returned.
00637                 emptyReturned = true;
00638                 /*System.out.println("Empty token for empty string.");*/
00639                 return true;
00640             } else {
00641                 char c = text.charAt(position);
00642                 if (c <= maxDelimChar &&
00643                     (nontokenDelims != null && nontokenDelims
00644                     .indexOf(c) != -1) ||
00645                     (tokenDelims != null && tokenDelims.indexOf(c) != -1)){
00646                     // There is a delimiter at the very start of the string
00647                     // so we must return an empty token at the beginning.
00648                     emptyReturned = true;
00649                     /*System.out.println("Empty token at beginning.");*/
00650                     return true;
00651                 }
00652             }
00653             // The main loop
00654             // Do this as long as parts of the string have yet to be examined
00655             while (position != -1){
00656                 char c = text.charAt(position);
00657                 if (returnEmptyTokens && !emptyReturned && position >
                    delimsChangedPosition){
00658                     char c1 = text.charAt(position - 1);
00659                     // Examine the current character and the one before it.
00660                     // If both of them are delimiters, then we need to

```

```

o return
00661                                     // an empty delimiter. Note that characters that
were examined
00662                                     // before the delimiters changed should not be re
examined.
00663                                     if (c <= maxDelimChar && c1 <= maxDelimChar &&
00664                                     ((nontokenDelims != null && nontokenDelim
s.indexOf(c) != -1) ||
00665                                     (tokenDelims != null && tokenDelims.index
Of(c) != -1)) &&
00666                                     ((nontokenDelims != null && nontokenDelim
s.indexOf(c1) != -1) ||
00667                                     (tokenDelims != null && tokenDelims.index
Of(c1) != -1))){
00668                                     emptyReturned = true;
00669                                     /*System.out.println("Empty token.");*/
00670                                     return true;
00671                                     }
00672                                     }
00673
00674                                     int nextDelimiter = (position < strLength - 1 ? indexOfNe
xtDelimiter(position + 1) : -1);
00675                                     if (c > maxDelimChar ||
00676                                     ((nontokenDelims == null || nontokenDelims.indexO
f(c) == -1) &&
00677                                     (tokenDelims == null || tokenDelims.indexOf(c) ==
-1))){
00678                                     // token found
00679                                     /*System.out.println("Token: '" +
00680                                     text.substring(position, (nextDelimiter =
= -1 ? strLength : nextDelimiter)) +
00681                                     "' at " + position + ".");*/
00682                                     position = nextDelimiter;
00683                                     emptyReturned = false;
00684                                     return true;
00685                                     } else if (tokenDelims != null && tokenDelims.indexOf(c)
!= -1) {
00686                                     // delimiter that can be returned as a token foun
d
00687                                     emptyReturned = false;
00688                                     /*System.out.println("Delimiter: '" + c + "' at "
+ position + ".");*/
00689                                     position = (position < strLength - 1 ? position + 1
: -1);
00690                                     return true;
00691                                     } else {
00692                                     // delimiter that is not a token found.
00693                                     emptyReturned = false;
00694                                     position = (position < strLength - 1 ? position + 1
: -1);
00695                                     return false;
00696                                     }
00697                                     }
00698                                     // handle the case that a token is at the end of the string and w
e should
00699                                     // return empty tokens.
00700                                     if (returnEmptyTokens && !emptyReturned && strLength > 0){

```

```

00701         char c = text.charAt(strLength - 1);
00702         if (c <= maxDelimChar &&
00703             (nontokenDelims != null && nontokenDelims.indexOf
00704             (c) != -1) ||
00705             (tokenDelims != null && tokenDelims.indexOf(c) !=
00706             -1)){
00707             // empty token at the end of the string found.
00708             emptyReturned = true;
00709             /*System.out.println("Empty token at end.");*/
00710             return true;
00711         }
00712     }
00713     return false;
00714 }
00715 public String nextToken(String nontokenDelims, String tokenDelims){
00716     setDelims(nontokenDelims, tokenDelims);
00717     return nextToken();
00718 }
00719 public String nextToken(String nontokenDelims, String tokenDelims, boolean
00720     returnEmptyTokens){
00721     setDelims(nontokenDelims, tokenDelims);
00722     setReturnEmptyTokens(returnEmptyTokens);
00723     return nextToken();
00724 }
00725 public String nextToken(String delims, boolean delimsAreTokens){
00726     return (delimsAreTokens ? nextToken(null, delims) : nextToken(del
00727     ims, null));
00728 }
00729 public String nextToken(String nontokenDelims){
00730     return nextToken(nontokenDelims, null);
00731 }
00732 private int indexOfNextDelimiter(int start){
00733     char c;
00734     int next;
00735     for (next = start; (c = text.charAt(next)) > maxDelimChar ||
00736         ((nontokenDelims == null || nontokenDelims.indexOf(c) ==
00737         -1) &&
00738         (tokenDelims == null || tokenDelims.indexOf(c) == -1)); n
00739     ext++){
00740         if (next == strLength - 1){
00741             // we have reached the end of the string without
00742             // finding a delimiter
00743             return (-1);
00744         }
00745     }
00746     return next;
00747 }
00748 public boolean hasMoreElements(){
00749     return hasMoreTokens();
00750 }
00751 }
00752 }
00753 }

```



```
00867     public Object nextElement() {
00868         return nextToken();
00869     }
00870
00882     public boolean hasNext() {
00883         return hasMoreTokens();
00884     }
00885
00899     public Object next() {
00900         return nextToken();
00901     }
00902
00912     public void remove() {
00913         throw new UnsupportedOperationException();
00914     }
00915
00937     public void setReturnEmptyTokens(boolean returnEmptyTokens) {
00938         // this could effect the number of tokens
00939         tokenCount = -1;
00940         this.returnEmptyTokens = returnEmptyTokens;
00941     }
00942
00952     public int getCurrentPosition() {
00953         return this.position;
00954     }
00955
00966     public String[] toArray() {
00967         String[] tokenArray = new String[countTokens()];
00968         for(int i=0; hasMoreTokens(); i++) {
00969             tokenArray[i] = nextToken();
00970         }
00971         return tokenArray;
00972     }
00973
00982     public String restOfText() {
00983         return nextToken(null, null);
00984     }
00985
00997     public String peek() {
00998         // copy over state variables from the class to local
00999         // variables so that the state of this object can be
01000         // restored to the state that it was in before this
01001         // method was called.
01002         int savedPosition = position;
01003         boolean savedEmptyReturned = emptyReturned;
01004         int savedtokenCount = tokenCount;
01005
01006         // get the next token
01007         String retval = nextToken();
01008
01009         // restore the state
01010         position = savedPosition;
01011         emptyReturned = savedEmptyReturned;
01012         tokenCount = savedtokenCount;
01013
01014         // return the nextToken;
01015         return(retval);
```

```
01016         }  
01017     }  
01018  
01019     class UnsupportedOperationException extends RuntimeException {  
01020     }  
01021 }
```


6.19 src/security/Sha1.java File Reference

Classes

- class [security.Sha1](#)

Packages

- package [security](#)

6.20 src/security/Sha1.java

```
00001 /*
00002  * To change this template, choose Tools | Templates
00003  * and open the template in the editor.
00004  */
00005
00006 package security;
00007
00008
00009
00010
00011 import java.security.DigestException;
00012 import java.security.MessageDigest;
00013 import java.security.NoSuchAlgorithmException;
00014
00015
00022 public class Sha1 {
00023
00024
00031     private MessageDigest md;
00032     private byte[] buffer = new byte[20];
00033     private byte[] digest = new byte[20];
00034     private String hash = "";
00035
00036     // MessageDigest md = MessageDigest.getInstance("SHA-1");
00037     // md.update(data, 0, data.length);
00038     // md.digest(digest, 0, 20);
00039
00040
00050     public String getHash(String message) throws NoSuchAlgorithmException {
00051         buffer = message.getBytes();
00052         md = MessageDigest.getInstance("SHA-1");
00053         md.update(buffer, 0, buffer.length);
00054         try {
00055             md.digest (digest, 0, digest.length);
00056         } catch (DigestException ex) {
00057             ex.printStackTrace();
00058         }
00059
00060         for(int i =0; i < digest.length; i++ ) {
00061             byte aux = digest[i];
00062             int b = aux & 0xff;
00063             if (Integer.toHexString(b).length() == 1) hash += "0";
00064             hash += Integer.toHexString(b);
00065         }
00066
00067         return hash;
00068     }
00069 }
```

Index

- beginParseFile
 - Parser::RestParser, 59
- beginParseFileContent
 - Parser::RestParser, 60
- beginParseFiles
 - Parser::RestParser, 60
- beginParseLicense
 - Parser::RestParser, 60, 61
- beginParseLicenses
 - Parser::RestParser, 61
- beginParsePropiedad
 - Parser::RestParser, 61, 62
- beginParseUser
 - Parser::RestParser, 62
- beginParseUsers
 - Parser::RestParser, 63
- commandAction
 - hello::HelloMIDlet, 24
- Core, 7
 - Core::File, 12
 - File, 12
 - getCreationdate, 13
 - getFilename, 13
 - getFilesize, 13
 - getId, 14
 - getLicenseCollection, 14
 - getLocalpath, 14
 - getModificationdate, 14
 - setCreationdate, 15
 - setFilename, 15
 - setFilesize, 15
 - setId, 15
 - setLicenseCollection, 15
 - setLocalpath, 16
 - setModificationdate, 16
 - toString, 16
 - writeFile, 16
 - Core::FileContent, 18
 - FileContent, 18
 - getContent, 19
 - getFileid, 19
 - getId, 19
 - setContent, 19
 - setFileid, 19
 - setId, 20
 - Core::License, 53
 - getFileid, 54
 - getId, 54
 - getType, 54
 - getUserid, 54
 - License, 53
 - setFileid, 54
 - setId, 55
 - setType, 55
 - setuserid, 55
 - toString, 55
 - Core::Propiedades, 57
 - getUrlserveridor, 57
 - Propiedades, 57
 - setUrlserveridor, 57
 - Core::User, 88
 - getId, 89
 - getLicenseCollection, 89
 - getPassword, 89
 - getUsername, 89
 - setId, 89
 - setLicenseCollection, 90
 - setPassword, 90
 - setUsername, 90
 - toString, 90
 - User, 88
- countTokens
 - Parser::StringTokenizer, 70–72

- delimsChangedPosition
 - Parser::StringTokenizer, 83
- destroyApp
 - hello::HelloMIDlet, 24
- emptyReturned
 - Parser::StringTokenizer, 84
- exitMIDlet
 - hello::HelloMIDlet, 25
- File
 - Core::File, 12
- FileContent
 - Core::FileContent, 18
- getAccesoCorrecto
 - hello::HelloMIDlet, 25
- getAccesoIncorrecto
 - hello::HelloMIDlet, 25
- getAlertaArchivos
 - hello::HelloMIDlet, 25
- getAlertaArchivosCS
 - hello::HelloMIDlet, 25
- getBackCommand
 - hello::HelloMIDlet, 26
- getBackCommand1
 - hello::HelloMIDlet, 26
- getBackCommand10
 - hello::HelloMIDlet, 26
- getBackCommand2
 - hello::HelloMIDlet, 26
- getBackCommand3
 - hello::HelloMIDlet, 27
- getBackCommand4
 - hello::HelloMIDlet, 27
- getBackCommand5
 - hello::HelloMIDlet, 27
- getBackCommand6
 - hello::HelloMIDlet, 27
- getBackCommand7
 - hello::HelloMIDlet, 27
- getBackCommand8
 - hello::HelloMIDlet, 28
- getBackCommand9
 - hello::HelloMIDlet, 28
- getBitacoraCreadaDown
 - hello::HelloMIDlet, 28
- getBitacoraCreadaUp
 - hello::HelloMIDlet, 28
- getBitacoraErrorDown
 - hello::HelloMIDlet, 29
- getBitacoraErrorUp
 - hello::HelloMIDlet, 29
- getCancelCommand
 - hello::HelloMIDlet, 29
- getCancelCommand1
 - hello::HelloMIDlet, 29
- getCancelCommand2
 - hello::HelloMIDlet, 29
- getCancelCommand25
 - hello::HelloMIDlet, 30
- getCancelCommand30
 - hello::HelloMIDlet, 30
- getChoiceGroup
 - hello::HelloMIDlet, 30
- getChoiceGroup1
 - hello::HelloMIDlet, 30
- getChoiceGroup2
 - hello::HelloMIDlet, 31
- getChoiceGroup3
 - hello::HelloMIDlet, 31
- getChoiceGroup4
 - hello::HelloMIDlet, 31
- getChoiceGroup5
 - hello::HelloMIDlet, 31
- getContent
 - Core::FileContent, 19
- getCreationdate
 - Core::File, 13
- getCurrentPosition
 - Parser::StringTokenizer, 73
- getDisplay
 - hello::HelloMIDlet, 31
- getDocument
 - Parser::RestParser, 63
- getDocumentLocal
 - Parser::RestParser, 63
- getExitCommand
 - hello::HelloMIDlet, 32
- getExitCommand1
 - hello::HelloMIDlet, 32
- getExitCommand2

- hello::HelloMIDlet, 32
- getFileBrowser
 - hello::HelloMIDlet, 32
- getFileid
 - Core::FileContent, 19
 - Core::License, 54
- getFilename
 - Core::File, 13
- getFileSize
 - Core::File, 13
- getForm
 - hello::HelloMIDlet, 33
- getForm1
 - hello::HelloMIDlet, 33
- getForm2
 - hello::HelloMIDlet, 33
- getForm3
 - hello::HelloMIDlet, 33
- getForm4
 - hello::HelloMIDlet, 33
- getForm5
 - hello::HelloMIDlet, 34
- getForm6
 - hello::HelloMIDlet, 34
- getHash
 - security::Sha1, 65
- getId
 - Core::File, 14
 - Core::FileContent, 19
 - Core::License, 54
 - Core::User, 89
- getImage1
 - hello::HelloMIDlet, 34
- getImage2
 - hello::HelloMIDlet, 34
- getImage3
 - hello::HelloMIDlet, 35
- getItemCommand
 - hello::HelloMIDlet, 35
- getItemCommand1
 - hello::HelloMIDlet, 35
- getItemCommand2
 - hello::HelloMIDlet, 35
- getItemCommand3
 - hello::HelloMIDlet, 35
- getItemCommand4
 - hello::HelloMIDlet, 36
- getItemCommand5
 - hello::HelloMIDlet, 36
- getLicenseCollection
 - Core::File, 14
 - Core::User, 89
- getList
 - hello::HelloMIDlet, 36
- getList2
 - hello::HelloMIDlet, 36
- getList3
 - hello::HelloMIDlet, 37
- getLccalpath
 - Core::File, 14
- getLcginScreen
 - hello::HelloMIDlet, 37
- getModificationdate
 - Core::File, 14
- getOkCommand
 - hello::HelloMIDlet, 37
- getOkCommand1
 - hello::HelloMIDlet, 37
- getOkCommand10
 - hello::HelloMIDlet, 37
- getOkCommand11
 - hello::HelloMIDlet, 38
- getOkCommand12
 - hello::HelloMIDlet, 38
- getOkCommand13
 - hello::HelloMIDlet, 38
- getOkCommand14
 - hello::HelloMIDlet, 38
- getOkCommand15
 - hello::HelloMIDlet, 39
- getOkCommand16
 - hello::HelloMIDlet, 39
- getOkCommand17
 - hello::HelloMIDlet, 39
- getOkCommand18
 - hello::HelloMIDlet, 39
- getOkCommand19
 - hello::HelloMIDlet, 39
- getOkCommand2
 - hello::HelloMIDlet, 40
- getOkCommand20
 - hello::HelloMIDlet, 40

- getOkCommand21
 - hello::HelloMIDlet, 40
- getOkCommand22
 - hello::HelloMIDlet, 40
- getOkCommand23
 - hello::HelloMIDlet, 41
- getOkCommand24
 - hello::HelloMIDlet, 41
- getOkCommand25
 - hello::HelloMIDlet, 41
- getOkCommand3
 - hello::HelloMIDlet, 41
- getOkCommand30
 - hello::HelloMIDlet, 41
- getOkCommand4
 - hello::HelloMIDlet, 42
- getOkCommand5
 - hello::HelloMIDlet, 42
- getOkCommand6
 - hello::HelloMIDlet, 42
- getOkCommand7
 - hello::HelloMIDlet, 42
- getOkCommand8
 - hello::HelloMIDlet, 43
- getOkCommand9
 - hello::HelloMIDlet, 43
- getPassword
 - Core::User, 89
- getProcesoAcceso
 - hello::HelloMIDlet, 43
- getSincCorrecto
 - hello::HelloMIDlet, 43
- getSincError
 - hello::HelloMIDlet, 43
- getSplashScreen
 - hello::HelloMIDlet, 44
- getTask
 - hello::HelloMIDlet, 44
- getTask1
 - hello::HelloMIDlet, 44
- getTask2
 - hello::HelloMIDlet, 44
- getTask3
 - hello::HelloMIDlet, 45
- getTask4
 - hello::HelloMIDlet, 45
- getTask5
 - hello::HelloMIDlet, 45
- getTask6
 - hello::HelloMIDlet, 45
- getTask7
 - hello::HelloMIDlet, 45
- getTextBox
 - hello::HelloMIDlet, 46
- getTextBox1
 - hello::HelloMIDlet, 46
- getTextBox5
 - hello::HelloMIDlet, 46
- getTextBox6
 - hello::HelloMIDlet, 46
- getTextField
 - hello::HelloMIDlet, 47
- getTicker
 - hello::HelloMIDlet, 47
- getTickerSalida
 - hello::HelloMIDlet, 47
- getTransferenciaCompletaDown
 - hello::HelloMIDlet, 47
- getTransferenciaCompletaUp
 - hello::HelloMIDlet, 47
- getTransferenciaErrorDown
 - hello::HelloMIDlet, 47
- getTransferenciaErrorUp
 - hello::HelloMIDlet, 48
- getType
 - Core::License, 54
- getUrlServidor
 - Core::Propiedades, 57
- getUserid
 - Core::License, 54
- getUsername
 - Core::User, 89
- getVerArchivosMovil
 - hello::HelloMIDlet, 48
- getVerArchivosServidor
 - hello::HelloMIDlet, 48
- getWaitScreen
 - hello::HelloMIDlet, 48
- getWaitScreen1
 - hello::HelloMIDlet, 49
- getWaitScreen2
 - hello::HelloMIDlet, 49

- getWaitScreen3
 - hello::HelloMIDlet, 49
- getWaitScreen4
 - hello::HelloMIDlet, 49
- getWaitScreen5
 - hello::HelloMIDlet, 49
- getWaitScreen6
 - hello::HelloMIDlet, 50
- hasMoreElements
 - Parser::StringTokenizer, 73
- hasMoreTokens
 - Parser::StringTokenizer, 73
- hasNext
 - Parser::StringTokenizer, 74
- hello, 8
- hello::HelloMIDlet, 21
 - commandAction, 24
 - destroyApp, 24
 - exitMIDlet, 25
 - getAccesoCorrecto, 25
 - getAccesoIncorrecto, 25
 - getAlertaArchivos, 25
 - getAlertaArchivosCS, 25
 - getBackCommand, 26
 - getBackCommand1, 26
 - getBackCommand10, 26
 - getBackCommand2, 26
 - getBackCommand3, 27
 - getBackCommand4, 27
 - getBackCommand5, 27
 - getBackCommand6, 27
 - getBackCommand7, 27
 - getBackCommand8, 28
 - getBackCommand9, 28
 - getBitacoraCreadaDown, 28
 - getBitacoraCreadaUp, 28
 - getBitacoraErrorDown, 29
 - getBitacoraErrorUp, 29
 - getCancelCommand, 29
 - getCancelCommand1, 29
 - getCancelCommand2, 29
 - getCancelCommand25, 30
 - getCancelCommand30, 30
 - getChoiceGroup, 30
 - getChoiceGroup1, 30
 - getChoiceGroup2, 31
 - getChoiceGroup3, 31
 - getChoiceGroup4, 31
 - getChoiceGroup5, 31
 - getDisplay, 31
 - getExitCommand, 32
 - getExitCommand1, 32
 - getExitCommand2, 32
 - getFileBrowser, 32
 - getForm, 33
 - getForm1, 33
 - getForm2, 33
 - getForm3, 33
 - getForm4, 33
 - getForm5, 34
 - getForm6, 34
 - getImage1, 34
 - getImage2, 34
 - getImage3, 35
 - getItemCommand, 35
 - getItemCommand1, 35
 - getItemCommand2, 35
 - getItemCommand3, 35
 - getItemCommand4, 36
 - getItemCommand5, 36
 - getList, 36
 - getList2, 36
 - getList3, 37
 - getLoginScreen, 37
 - getOkCommand, 37
 - getOkCommand1, 37
 - getOkCommand10, 37
 - getOkCommand11, 38
 - getOkCommand12, 38
 - getOkCommand13, 38
 - getOkCommand14, 38
 - getOkCommand15, 39
 - getOkCommand16, 39
 - getOkCommand17, 39
 - getOkCommand18, 39
 - getOkCommand19, 39
 - getOkCommand2, 40
 - getOkCommand20, 40
 - getOkCommand21, 40
 - getOkCommand22, 40
 - getOkCommand23, 41

- getOkCommand24, 41
- getOkCommand25, 41
- getOkCommand3, 41
- getOkCommand30, 41
- getOkCommand4, 42
- getOkCommand5, 42
- getOkCommand6, 42
- getOkCommand7, 42
- getOkCommand8, 43
- getOkCommand9, 43
- getProcesoAcceso, 43
- getSincCorrecto, 43
- getSincError, 43
- getSplashScreen, 44
- getTask, 44
- getTask1, 44
- getTask2, 44
- getTask3, 45
- getTask4, 45
- getTask5, 45
- getTask6, 45
- getTask7, 45
- getTextBox, 46
- getTextBox1, 46
- getTextBox5, 46
- getTextBox6, 46
- getTextField, 47
- getTicker, 47
- getTickerSalida, 47
- getTransferenciaCompletaDown, 47
- getTransferenciaCompletaUp, 47
- getTransferenciaErrorDown, 47
- getTransferenciaErrorUp, 48
- getVerArchivosMovil, 48
- getVerArchivosServidor, 48
- getWaitScreen, 48
- getWaitScreen1, 49
- getWaitScreen2, 49
- getWaitScreen3, 49
- getWaitScreen4, 49
- getWaitScreen5, 49
- getWaitScreen6, 50
- HelloMIDlet, 24
- list2Action, 50
- list3Action, 50
- listAction, 50
- method, 50
- method1, 50
- pauseApp, 51
- readLine, 51
- resumeMIDlet, 51
- startApp, 51
- startMIDlet, 51
- switchDisplayable, 51
- HelloMIDlet
 - hello::HelloMIDlet, 24
- License
 - Core::License, 53
- list2Action
 - hello::HelloMIDlet, 50
- list3Action
 - hello::HelloMIDlet, 50
- listAction
 - hello::HelloMIDlet, 50
- maxDelimChar
 - Parser::StringTokenizer, 84
- method
 - hello::HelloMIDlet, 50
- method1
 - hello::HelloMIDlet, 50
- next
 - Parser::StringTokenizer, 74
- nextElement
 - Parser::StringTokenizer, 75
- nextToken
 - Parser::StringTokenizer, 75–78
- nontokenDelims
 - Parser::StringTokenizer, 84
- Parser, 9
- Parser::ConectorWeb, 11
 - sendHttpGet, 11
- Parser::RestParser, 59
 - beginParseFile, 59
 - beginParseFileContent, 60
 - beginParseFiles, 60
 - beginParseLicense, 60, 61
 - beginParseLicenses, 61
 - beginParsePropiedad, 61, 62

- beginParseUser, 62
- beginParseUsers, 63
- getDocument, 63
- getDocumentLocal, 63
- setDocumentLocal, 63
- Parser::StringTokenizer, 66
 - countTokens, 70–72
 - delimsChangedPosition, 83
 - emptyReturned, 84
 - getCurrentPosition, 73
 - hasMoreElements, 73
 - hasMoreTokens, 73
 - hasNext, 74
 - maxDelimChar, 84
 - next, 74
 - nextElement, 75
 - nextToken, 75–78
 - nontokenDelims, 84
 - peek, 78
 - position, 84
 - remove, 79
 - restOfText, 79
 - returnEmptyTokens, 85
 - setDelimiters, 79–81
 - setReturnEmptyTokens, 81
 - setText, 82
 - skipDelimiters, 83
 - StringTokenizer, 67–69
 - strLength, 85
 - text, 85
 - toArray, 83
 - tokenCount, 85
 - tokenDelims, 86
- Parser::UnsupportedOperationException, 87
- pauseApp
 - hello::HelloMIDlet, 51
- peek
 - Parser::StringTokenizer, 78
- position
 - Parser::StringTokenizer, 84
- Propiedades
 - Core::Propiedades, 57
- readLine
 - hello::HelloMIDlet, 51
- remove
 - Parser::StringTokenizer, 79
- restOfText
 - Parser::StringTokenizer, 79
- resumeMIDlet
 - hello::HelloMIDlet, 51
- returnEmptyTokens
 - Parser::StringTokenizer, 85
- security, 10
- security::Sha1, 65
 - getHash, 65
- sendHttpGet
 - Parser::ConectorWeb, 11
- setContent
 - Core::FileContent, 19
- setCreationdate
 - Core::File, 15
- setDelimiters
 - Parser::StringTokenizer, 79–81
- setDocumentLocal
 - Parser::RestParser, 63
- setFileid
 - Core::FileContent, 19
 - Core::License, 54
- setFilename
 - Core::File, 15
- setFilesize
 - Core::File, 15
- setId
 - Core::File, 15
 - Core::FileContent, 20
 - Core::License, 55
 - Core::User, 89
- setLicenseCollection
 - Core::File, 15
 - Core::User, 90
- setLocalpath
 - Core::File, 16
- setModificationdate
 - Core::File, 16
- setPassword
 - Core::User, 90
- setReturnEmptyTokens
 - Parser::StringTokenizer, 81
- setText

Parser::StringTokenizer, 82
 setType
 Core::License, 55
 setUrlservidor
 Core::Propiedades, 57
 setUserid
 Core::License, 55
 setUsername
 Core::User, 90
 skipDelimiters
 Parser::StringTokenizer, 83
 src/Core/File.java, 93
 src/Core/FileContent.java, 97
 src/Core/License.java, 99
 src/Core/Propiedades.java, 102
 src/Core/User.java, 104
 src/hello/HelloMIDlet.java, 107
 src/Parser/ConectorWeb.java, 192
 src/Parser/RestParser.java, 194
 src/Parser/StringTokenizer.java, 204
 src/security/Sha1.java, 215
 startApp
 hello::HelloMIDlet, 51
 startMIDlet
 hello::HelloMIDlet, 51
 StringTokenizer
 Parser::StringTokenizer, 67–69
 strLength
 Parser::StringTokenizer, 85
 switchDisplayable
 hello::HelloMIDlet, 51

 text
 Parser::StringTokenizer, 85
 toArray
 Parser::StringTokenizer, 83
 tokenCount
 Parser::StringTokenizer, 85
 tokenDelims
 Parser::StringTokenizer, 86
 toString
 Core::File, 16
 Core::License, 55
 Core::User, 90

 User
 Core::User, 88
 writeFile
 Core::File, 16

XMLTest
1.0

Generated by Doxygen 1.6.3

Mon Apr 26 17:09:39 2010

Contents

1	Namespace Index	1
1.1	Package List	1
2	Class Index	3
2.1	Class List	3
3	File Index	5
3.1	File List	5
4	Namespace Documentation	7
4.1	Package acceso	7
4.2	Package archivodate	8
4.3	Package compara	9
4.4	Package comparaup	10
4.5	Package crea	11
4.6	Package obtiearchivo	12
4.7	Package reemplazo	13
4.8	Package regresa	14
4.9	Package valor	15
4.10	Package verifica	16
4.11	Package visualiza	17
5	Class Documentation	19
5.1	obtiearchivo.archivo Class Reference	19

5.1.1	Detailed Description	19
5.1.2	Member Function Documentation	19
5.1.2.1	descarga	19
5.2	obtienearchivo.archivoPort Class Reference	21
5.2.1	Detailed Description	21
5.2.2	Constructor & Destructor Documentation	21
5.2.2.1	archivoPort	21
5.2.3	Member Function Documentation	21
5.2.3.1	getDescarga	21
5.3	reemplazo.archivosreemplazo Class Reference	23
5.3.1	Detailed Description	23
5.3.2	Member Function Documentation	23
5.3.2.1	lista	23
5.4	reemplazo.archivosreemplazoPort Class Reference	24
5.4.1	Detailed Description	24
5.4.2	Constructor & Destructor Documentation	24
5.4.2.1	archivosreemplazoPort	24
5.4.3	Member Function Documentation	24
5.4.3.1	getListas	24
5.5	compara.bitacoras Class Reference	26
5.5.1	Detailed Description	26
5.5.2	Member Function Documentation	26
5.5.2.1	comparabitacoras	26
5.6	compara.bitacorasPort Class Reference	27
5.6.1	Detailed Description	27
5.6.2	Constructor & Destructor Documentation	27
5.6.2.1	bitacorasPort	27
5.6.3	Member Function Documentation	27
5.6.3.1	getComparabitacoras	27
5.7	comparaup.comparabitacoraup Class Reference	29
5.7.1	Detailed Description	29

5.7.2	Member Function Documentation	29
5.7.2.1	comparabitacoras	29
5.8	comparaup.comparabitacoraupPort Class Reference	30
5.8.1	Detailed Description	30
5.8.2	Constructor & Destructor Documentation	30
5.8.2.1	comparabitacoraupPort	30
5.8.3	Member Function Documentation	30
5.8.3.1	getComparabitacoras	30
5.9	regresa.espejo Class Reference	32
5.9.1	Detailed Description	32
5.9.2	Member Function Documentation	32
5.9.2.1	igual	32
5.10	regresa.espejoPort Class Reference	33
5.10.1	Detailed Description	33
5.10.2	Constructor & Destructor Documentation	33
5.10.2.1	espejoPort	33
5.10.3	Member Function Documentation	33
5.10.3.1	getIgual	33
5.11	verifica.existe Class Reference	34
5.11.1	Detailed Description	34
5.11.2	Member Function Documentation	34
5.11.2.1	verificaarchivo	34
5.12	verifica.existePort Class Reference	35
5.12.1	Detailed Description	35
5.12.2	Constructor & Destructor Documentation	35
5.12.2.1	existePort	35
5.12.3	Member Function Documentation	35
5.12.3.1	getVerificaarchivo	35
5.13	archivodate.fecha Class Reference	37
5.13.1	Detailed Description	37
5.13.2	Member Function Documentation	37

5.13.2.1	fechaedicion	37
5.14	archivodate.fechaPort Class Reference	38
5.14.1	Detailed Description	38
5.14.2	Constructor & Destructor Documentation	38
5.14.2.1	fechaPort	38
5.14.3	Member Function Documentation	38
5.14.3.1	getFechaedicion	38
5.15	crea.recibe Class Reference	40
5.15.1	Detailed Description	40
5.15.2	Member Function Documentation	40
5.15.2.1	creararchivo	40
5.16	crea.recibePort Class Reference	41
5.16.1	Detailed Description	41
5.16.2	Constructor & Destructor Documentation	41
5.16.2.1	recibePort	41
5.16.3	Member Function Documentation	41
5.16.3.1	getCreararchivo	41
5.17	valor.tamano Class Reference	43
5.17.1	Detailed Description	43
5.17.2	Member Function Documentation	43
5.17.2.1	archivokb	43
5.18	valor.tamanoPort Class Reference	44
5.18.1	Detailed Description	44
5.18.2	Constructor & Destructor Documentation	44
5.18.2.1	tamanoPort	44
5.18.3	Member Function Documentation	44
5.18.3.1	getArchivokb	44
5.19	visualiza.todosarchivos Class Reference	46
5.19.1	Detailed Description	46
5.19.2	Member Function Documentation	46
5.19.2.1	lista	46

5.20	visualiza.todosarchivosPort Class Reference	47
5.20.1	Detailed Description	47
5.20.2	Constructor & Destructor Documentation	47
5.20.2.1	todosarchivosPort	47
5.20.3	Member Function Documentation	47
5.20.3.1	getLista	47
5.21	acceso.user Class Reference	49
5.21.1	Detailed Description	49
5.21.2	Member Function Documentation	49
5.21.2.1	valida	49
5.22	acceso.userPort Class Reference	50
5.22.1	Detailed Description	50
5.22.2	Constructor & Destructor Documentation	50
5.22.2.1	userPort	50
5.22.3	Member Function Documentation	50
5.22.3.1	getValida	50
6	File Documentation	53
6.1	src/java/acceso/user.java File Reference	53
6.2	src/java/acceso/user.java	54
6.3	src/java/acceso/userPort.java File Reference	55
6.4	src/java/acceso/userPort.java	56
6.5	src/java/archivodate/fecha.java File Reference	58
6.6	src/java/archivodate/fecha.java	59
6.7	src/java/archivodate/fechaPort.java File Reference	61
6.8	src/java/archivodate/fechaPort.java	62
6.9	src/java/compara/bitacoras.java File Reference	64
6.10	src/java/compara/bitacoras.java	65
6.11	src/java/compara/bitacorasPort.java File Reference	67
6.12	src/java/compara/bitacorasPort.java	68
6.13	src/java/comparaup/comparabitacoraup.java File Reference	70

6.14	src/java/comparaup/comparabitacoraup.java	71
6.15	src/java/comparaup/comparabitacoraupPort.java File Reference	73
6.16	src/java/comparaup/comparabitacoraupPort.java	74
6.17	src/java/crea/recibe.java File Reference	76
6.18	src/java/crea/recibe.java	77
6.19	src/java/crea/recibePort.java File Reference	79
6.20	src/java/crea/recibePort.java	80
6.21	src/java/obtienearchivo/archivo.java File Reference	82
6.22	src/java/obtienearchivo/archivo.java	83
6.23	src/java/obtienearchivo/archivoPort.java File Reference	85
6.24	src/java/obtienearchivo/archivoPort.java	86
6.25	src/java/reemplazo/archivosreemplazo.java File Reference	88
6.26	src/java/reemplazo/archivosreemplazo.java	89
6.27	src/java/reemplazo/archivosreemplazoPort.java File Reference	91
6.28	src/java/reemplazo/archivosreemplazoPort.java	92
6.29	src/java/regresa/espejo.java File Reference	94
6.30	src/java/regresa/espejo.java	95
6.31	src/java/regresa/espejoPort.java File Reference	96
6.32	src/java/regresa/espejoPort.java	97
6.33	src/java/valor/tamano.java File Reference	99
6.34	src/java/valor/tamano.java	100
6.35	src/java/valor/tamanoPort.java File Reference	102
6.36	src/java/valor/tamanoPort.java	103
6.37	src/java/verifica/existe.java File Reference	105
6.38	src/java/verifica/existe.java	106
6.39	src/java/verifica/existePort.java File Reference	108
6.40	src/java/verifica/existePort.java	109
6.41	src/java/visualiza/todosarchivos.java File Reference	111
6.42	src/java/visualiza/todosarchivos.java	112
6.43	src/java/visualiza/todosarchivosPort.java File Reference	114
6.44	src/java/visualiza/todosarchivosPort.java	115

Chapter 1

Namespace Index

1.1 Package List

Here are the packages with brief descriptions (if available):

acceso	7
archivodate	8
compara	9
comparaup	10
crea	11
obtienearchivo	12
reemplazo	13
regresa	14
valor	15
verifica	16
visualiza	17

Chapter 2

Class Index

2.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

obtienearchivo.archivo	19
obtienearchivo.archivoPort	21
reemplazo.archivosreemplazo	23
reemplazo.archivosreemplazoPort	24
compara.bitacoras	26
compara.bitacorasPort	27
comparaup.comparabitacoraup	29
comparaup.comparabitacoraupPort	30
regresa.espejo	32
regresa.espejoPort	33
verifica.existe	34
verifica.existePort	35
archivodate.fecha	37
archivodate.fechaPort	38
crea.recibe	40
crea.recibePort	41
valor.tamano	43
valor.tamanoPort	44
visualiza.todosarchivos	46
visualiza.todosarchivosPort	47
acceso.user	49
acceso.userPort	50

Chapter 3

File Index

3.1 File List

Here is a list of all files with brief descriptions:

src/java/acceso/user.java	53
src/java/acceso/userPort.java	55
src/java/archivodate/fecha.java	58
src/java/archivodate/fechaPort.java	61
src/java/compara/bitacoras.java	64
src/java/compara/bitacorasPort.java	67
src/java/comparaup/comparabitacoraup.java	70
src/java/comparaup/comparabitacoraupPort.java	73
src/java/crea/recibe.java	76
src/java/crea/recibePort.java	79
src/java/obtienearchivo/archivo.java	82
src/java/obtienearchivo/archivoPort.java	85
src/java/reemplazo/archivosreemplazo.java	88
src/java/reemplazo/archivosreemplazoPort.java	91
src/java/regresa/espejo.java	94
src/java/regresa/espejoPort.java	96
src/java/valor/tamano.java	99
src/java/valor/tamanoPort.java	102
src/java/verifica/existe.java	105
src/java/verifica/existePort.java	108
src/java/visualiza/todosarchivos.java	111
src/java/visualiza/todosarchivosPort.java	114

Chapter 4

Namespace Documentation

4.1 Package acceso

Classes

- class `user`
- class `userPort`

4.2 Package archivodate

Classes

- class [fecha](#)
- class [fechaPort](#)

4.3 Package compara

Classes

- class [bitacoras](#)
- class [bitacorasPort](#)

4.4 Package comparaup

Classes

- class [comparabitacoraup](#)
- class [comparabitacoraupPort](#)

4.5 Package crea

Classes

- class [recibe](#)
- class [recibePort](#)

4.6 Package obtiencarchivo

Classes

- class [archivo](#)
- class [archivoPort](#)

4.7 Package reemplazo

Classes

- class [archivosreemplazo](#)
- class [archivosreemplazoPort](#)

4.8 Package `regresa`

Classes

- class `espejo`
- class `espejoPort`

4.9 Package valor

Classes

- class [tamanho](#)
- class [tamanhoPort](#)

4.10 Package verifica

Classes

- class [existe](#)
- class [existePort](#)

4.11 Package visualiza

Classes

- class [todosarchivos](#)
- class [todosarchivosPort](#)

Chapter 5

Class Documentation

5.1 obtienearchivo.archivo Class Reference

Public Member Functions

- String `descarga` (@WebParam(name="nombrearchivo") String nombrearchivo, @WebParam(name="usuario") String usuario, @WebParam(name="password") String password)

5.1.1 Detailed Description

Author

Rodrigo Ramírez Reyes

Definition at line 29 of file `archivo.java`.

5.1.2 Member Function Documentation

- ##### 5.1.2.1 String obtienearchivo.archivo.descarga
- (@WebParam(name="nombrearchivo") String *nombrearchivo*,
@WebParam(name="usuario") String *usuario*,
@WebParam(name="password") String *password*)

Web service operation Regresa el contenido de un archivo si se encuentra en el servidor

Parameters

nombrearchivo Nombre del archivo solicitado

usuario Nombre de usuario que solicita el servicio

password Password cifrado del usuario que solicita el servicio

Returns

Regresa el contenido del archivo solicitado

Definition at line 44 of file [archivo.java](#).

The documentation for this class was generated from the following file:

- [src/java/obtienearchivo/archivo.java](#)

5.2 obtienearchivo.archivoPort Class Reference

Public Member Functions

- [archivoPort \(\)](#)
- [String getDescarga \(@QueryParam\("nombreactivo"\) String nombreactivo, @QueryParam\("usuario"\) String usuario, @QueryParam\("password"\) String password\)](#)

5.2.1 Detailed Description

REST Web Service

Author

Rodrigo

Definition at line 23 of file [archivoPort.java](#).

5.2.2 Constructor & Destructor Documentation

5.2.2.1 obtienearchivo.archivoPort.archivoPort ()

Creates a new instance of [archivoPort](#)

Definition at line 29 of file [archivoPort.java](#).

5.2.3 Member Function Documentation

5.2.3.1 String obtienearchivo.archivoPort.getDescarga (@QueryParam("nombreactivo") String nombreactivo, @QueryParam("usuario") String usuario, @QueryParam("password") String password)

Invokes the SOAP method descarga

Parameters

nombreactivo resource URI parameter

Returns

an instance of java.lang.String

Definition at line 42 of file [archivoPort.java](#).

The documentation for this class was generated from the following file:

- [src/java/obtienearchivo/archivoPort.java](#)

5.3 reemplazo.archivosreemplazo Class Reference

Public Member Functions

- String [lista](#) (@WebParam(name="usuario") String usuario,@WebParam(name="password") String password)

5.3.1 Detailed Description

Author

Rodrigo Ramírez Reyes

Definition at line 23 of file [archivosreemplazo.java](#).

5.3.2 Member Function Documentation

5.3.2.1 String reemplazo.archivosreemplazo.lista (@WebParam(name="usuario") String *usuario*, @WebParam(name="password") String *password*)

Web service operation Devuelve la lista completa de los archivos que se encuentran en el servidor.

Parameters

usuario Nombre de usuario que solicita el servicio

password Password cifrado del usuario que solicita el servicio

Returns

Listado de archivos del servidor

Definition at line 38 of file [archivosreemplazo.java](#).

The documentation for this class was generated from the following file:

- [src/java/reemplazo/archivosreemplazo.java](#)

5.4 reemplazo.archivosreemplazoPort Class Reference

Public Member Functions

- [archivosreemplazoPort \(\)](#)
- [String archivosreemplazoPort.getLista \(@QueryParam\("usuario"\) String usuario, @QueryParam\("password"\) String password\)](#) String

5.4.1 Detailed Description

REST Web Service

Author

Rodrigo

Definition at line 24 of file [archivosreemplazoPort.java](#).

5.4.2 Constructor & Destructor Documentation

5.4.2.1 reemplazo.archivosreemplazoPort.archivosreemplazoPort ()

Creates a new instance of [archivosreemplazoPort](#)

Definition at line 30 of file [archivosreemplazoPort.java](#).

5.4.3 Member Function Documentation

5.4.3.1 String reemplazo.archivosreemplazoPort.getLista (@QueryParam("usuario") String usuario, @QueryParam("password") String password)

Invokes the SOAP method lista

Parameters

usuario resource URI parameter

Returns

an instance of `java.lang.String`

Definition at line 43 of file [archivosreemplazoPort.java](#).

The documentation for this class was generated from the following file:

- [src/java/reemplazo/archivosreemplazoPort.java](#)

5.5 compara.bitacoras Class Reference

Public Member Functions

- String `comparabitacoras` (`@WebParam(name="arbolArchivos")`
String `arbolArchivos`, `@WebParam(name="usuario")` String
`usuario`, `@WebParam(name="password")` String `password`)

5.5.1 Detailed Description

Author

Rodrigo Ramírez Reyes

Definition at line 25 of file `bitacoras.java`.

5.5.2 Member Function Documentation

- #### 5.5.2.1 String `compara.bitacoras.comparabitacoras` (`@WebParam(name="arbolArchivos")` String `arbolArchivos`, `@WebParam(name="usuario")` String `usuario`, `@WebParam(name="password")` String `password`)

Web service operation Realiza una comparación entre el directorio de archivos del cliente y el directorio de archivos del servidor para realizar descarga de archivos nuevos.

Parameters

arbolArchivos Listado del directorio de archivos del cliente
usuario Nombre de usuario que solicita el servicio
password Password cifrado del usuario que solicita el servicio

Returns

Regresa la comparación de directorios de archivos en modo Descarga

Definition at line 42 of file `bitacoras.java`.

The documentation for this class was generated from the following file:

- `src/java/compara/bitacoras.java`

5.6 compara.bitacorasPort Class Reference

Public Member Functions

- [bitacorasPort \(\)](#)
- [String getComparabitacoras \(@QueryParam\("arbolArchivos"\) String arbolArchivos,@QueryParam\("usuario"\) String usuario,@QueryParam\("password"\) String password\)](#)

5.6.1 Detailed Description

REST Web Service

Author

Rodrigo

Definition at line 23 of file [bitacorasPort.java](#).

5.6.2 Constructor & Destructor Documentation

5.6.2.1 [compara.bitacorasPort.bitacorasPort \(\)](#)

Creates a new instance of [bitacorasPort](#)

Definition at line 29 of file [bitacorasPort.java](#).

5.6.3 Member Function Documentation

5.6.3.1 [String compara.bitacorasPort.getComparabitacoras \(@QueryParam\("arbolArchivos"\) String arbolArchivos, @QueryParam\("usuario"\) String usuario, @QueryParam\("password"\) String password\)](#)

Invokes the SOAP method `comparabitacoras`

Parameters

arbolArchivos resource URI parameter

Returns

an instance of `java.lang.String`

Definition at line 42 of file [bitacorasPort.java](#).

The documentation for this class was generated from the following file:

- [src/java/compara/bitacorasPort.java](#)

5.7 comparaup.comparabitacoraup Class Reference

Public Member Functions

- String `comparabitacoras` (`@WebParam(name="arbolArchivos")`
String `arbolArchivos`, `@WebParam(name="usuario")` String
`usuario`, `@WebParam(name="password")` String `password`)

5.7.1 Detailed Description

Author

Rodrigo Ramírez Reyes

Definition at line 25 of file `comparabitacoraup.java`.

5.7.2 Member Function Documentation

- 5.7.2.1 String `comparaup.comparabitacoraup.comparabitacoras`
(`@WebParam(name="arbolArchivos")` String `arbolArchivos`,
`@WebParam(name="usuario")` String `usuario`,
`@WebParam(name="password")` String `password`)

Web service operation Realiza una comparación entre el directorio de archivos del cliente y el directorio de archivos del servidor para realizar carga de archivos nuevos.

Parameters

- arbolArchivos* Listado del directorio de archivos del cliente
- usuario* Nombre de usuario que solicita el servicio
- password* Password cifrado del usuario que solicita el servicio

Returns

Regresa la comparación de directorios de archivos en modo Carga

Definition at line 42 of file `comparabitacoraup.java`.

The documentation for this class was generated from the following file:

- `src/java/comparaup/comparabitacoraup.java`

5.8 comparaup.comparabitacoraupPort Class Reference

Public Member Functions

- `comparabitacoraupPort ()`
- `String getComparabitacoras (@QueryParam("arbolArchivos") String arbolArchivos, @QueryParam("usuario") String usuario, @QueryParam("password") String password)`

5.8.1 Detailed Description

REST Web Service

Author

Rodrigo

Definition at line 23 of file `comparabitacoraupPort.java`.

5.8.2 Constructor & Destructor Documentation

5.8.2.1 `comparaup.comparabitacoraupPort.comparabitacoraupPort ()`

Creates a new instance of `comparabitacoraupPort`

Definition at line 29 of file `comparabitacoraupPort.java`.

5.8.3 Member Function Documentation

5.8.3.1 `String comparaup.comparabitacoraupPort.getComparabitacoras (@QueryParam("arbolArchivos") String arbolArchivos, @QueryParam("usuario") String usuario, @QueryParam("password") String password)`

Invokes the SOAP method `comparabitacoras`

Parameters

arbolArchivos resource URI parameter

Returns

an instance of `java.lang.String`

Definition at line 42 of file `comparabitacoraupPort.java`.

The documentation for this class was generated from the following file:

- `src/java/comparaup/comparabitacoraupPort.java`

5.9 `regresa.espejo` Class Reference

Public Member Functions

- `String igual (@WebParam(name="arbolarchivos") String arbolarchivos)`

5.9.1 Detailed Description

Author

Rodrigo Ramírez Reyes

Definition at line 18 of file `espejo.java`.

5.9.2 Member Function Documentation

5.9.2.1 `String regresa.espejo.igual (@WebParam(name="arbolarchivos") String arbolarchivos)`

Web service operation Regresa el mismo listado del directorio de archivos del cliente con el formato para transferir archivos.

Parameters

arbolarchivos Listado de archivos del directorio del cliente

Returns

Regresa el directorio para la lectura de archivos en modo reemplazo

Definition at line 32 of file `espejo.java`.

The documentation for this class was generated from the following file:

- `src/java/regresa/espejo.java`

5.10 `regresa.espejoPort` Class Reference

Public Member Functions

- `espejoPort ()`
- `String getIgual (@QueryParam("arbolarchivos") String arbolarchivos)`

5.10.1 Detailed Description

REST Web Service

Author

Rodrigo

Definition at line 23 of file `espejoPort.java`.

5.10.2 Constructor & Destructor Documentation

5.10.2.1 `regresa.espejoPort.espejoPort ()`

Creates a new instance of `espejoPort`

Definition at line 29 of file `espejoPort.java`.

5.10.3 Member Function Documentation

5.10.3.1 `String regresa.espejoPort.getIgual (@QueryParam("arbolarchivos") String arbolarchivos)`

Invokes the SOAP method `igual`

Parameters

arbolarchivos resource URI parameter

Returns

an instance of `java.lang.String`

Definition at line 42 of file `espejoPort.java`.

The documentation for this class was generated from the following file:

- `src/java/regresa/espejoPort.java`

5.11 verifica.existe Class Reference

Public Member Functions

- String `verificaarchivo` (`@WebParam(name="nombre")`
String `nombre`, `@WebParam(name="usuario")` String
usuario, `@WebParam(name="password")` String `password`)

5.11.1 Detailed Description

Author

Rodrigo Ramírez Reyes

Definition at line 24 of file `existe.java`.

5.11.2 Member Function Documentation

- 5.11.2.1 String `verifica.existe.verificaarchivo` (`@WebParam(name="nombre")`
String `nombre`, `@WebParam(name="usuario")` String `usuario`,
`@WebParam(name="password")` String `password`)

Web service operation Verifica si el nombre de un archivo existe en el directorio de archivos del servidor.

Parameters

- nombre* Nombre de un archivo
- usuario* Nombre de usuario que solicita el servicio
- password* Password cifrado del usuario que solicita el servicio

Returns

Regresa true si el archivo solicitado existe en el servidor de archivos

Definition at line 40 of file `existe.java`.

The documentation for this class was generated from the following file:

- `src/java/verifica/existe.java`

5.12 verifica.existePort Class Reference

Public Member Functions

- `existePort ()`
- `String getVerificaarchivo (@QueryParam("nombre") String nombre, @QueryParam("usuario") String usuario, @QueryParam("password") String password)`

5.12.1 Detailed Description

REST Web Service

Author

Rodrigo

Definition at line 24 of file `existePort.java`.

5.12.2 Constructor & Destructor Documentation

5.12.2.1 verifica.existePort.existePort ()

Creates a new instance of `existePort`

Definition at line 30 of file `existePort.java`.

5.12.3 Member Function Documentation

5.12.3.1 String verifica.existePort.getVerificaarchivo (@QueryParam("nombre") String nombre, @QueryParam("usuario") String usuario, @QueryParam("password") String password)

Invokes the SOAP method `verificaarchivo`

Parameters

nombre resource URI parameter

Returns

an instance of `java.lang.String`

Definition at line 44 of file [existePort.java](#).

The documentation for this class was generated from the following file:

- [src/java/verifica/existePort.java](#)

5.13 archivodate.fecha Class Reference

Public Member Functions

- String [fechaedicion](#) (@WebParam(name="nombre") nombre, @WebParam(name="usuario") String usuario, @WebParam(name="password") String password)

5.13.1 Detailed Description

Author

Rodrigo Ramírez Reyes

Definition at line 25 of file [fecha.java](#).

5.13.2 Member Function Documentation

- 5.13.2.1 String [archivodate.fecha.fechaedicion](#) (@WebParam(name="nombre") String *nombre*, @WebParam(name="usuario") String *usuario*, @WebParam(name="password") String *password*)

Web service operation Regresa la última fecha de modificación de un archivo del directorio de archivos del servidor.

Parameters

- nombre* Nombre de un archivo
- usuario* Nombre de usuario que solicita el servicio
- password* Password cifrado del usuario que solicita el servicio

Returns

Regresa la ultima fecha de modificacion del archivo solicitado

Definition at line 41 of file [fecha.java](#).

The documentation for this class was generated from the following file:

- [src/java/archivodate/fecha.java](#)

5.14 archivodate.fechaPort Class Reference

Public Member Functions

- [fechaPort \(\)](#)
- String [getFechaedicion](#) (@QueryParam("nombre") String nombre, @QueryParam("usuario") String usuario, @QueryParam("password") String password)

5.14.1 Detailed Description

REST Web Service

Author

Rodrigo

Definition at line 23 of file [fechaPort.java](#).

5.14.2 Constructor & Destructor Documentation

5.14.2.1 archivodate.fechaPort.fechaPort ()

Creates a new instance of [fechaPort](#)

Definition at line 29 of file [fechaPort.java](#).

5.14.3 Member Function Documentation

5.14.3.1 String archivodate.fechaPort.getFechaedicion (@QueryParam("nombre") String nombre, @QueryParam("usuario") String usuario, @QueryParam("password") String password)

Invokes the SOAP method fechaedicion

Parameters

nombre resource URI parameter

usuario resource URI parameter

password resource URI parameter

Returns

an instance of java.lang.String

Definition at line 44 of file [fechaPort.java](#).

The documentation for this class was generated from the following file:

- [src/java/archivodate/fechaPort.java](#)

5.15 crea.recibe Class Reference

Public Member Functions

- String [creararchivo](#) (@WebParam(name="nombre") String nombre, @WebParam(name="contenido") String contenido, @WebParam(name="usuario") String usuario, @WebParam(name="password") String password)

5.15.1 Detailed Description

Author

Rodrigo Ramírez Reyes

Definition at line 30 of file [recibe.java](#).

5.15.2 Member Function Documentation

- 5.15.2.1** String [crea.recibe.creararchivo](#) (@WebParam(name="nombre") String *nombre*, @WebParam(name="contenido") String *contenido*, @WebParam(name="usuario") String *usuario*, @WebParam(name="password") String *password*)

Web service operation Crea un archivo en el servidor de archivos, con el nombre del archivo y el contenido que envía la aplicación cliente.

Parameters

- nombre* Nombre de un archivo
- contenido* Contenido de un archivo
- usuario* Nombre de usuario que solicita el servicio
- password* Password cifrado del usuario que solicita el servicio

Returns

Regresa true si el archivo pudo ser creado en el servidor de archivos

Definition at line 48 of file [recibe.java](#).

The documentation for this class was generated from the following file:

- [src/java/crea/recibe.java](#)

5.16 crea.recibePort Class Reference

Public Member Functions

- `recibePort ()`
- `String getCreararchivo (@QueryParam("nombre") String nombre, @QueryParam("contenido") String contenido, @QueryParam("usuario") String usuario, @QueryParam("password") String password)`

5.16.1 Detailed Description

REST Web Service

Author

Rodrigo

Definition at line 23 of file `recibePort.java`.

5.16.2 Constructor & Destructor Documentation

5.16.2.1 crea.recibePort.recibePort ()

Creates a new instance of `recibePort`

Definition at line 29 of file `recibePort.java`.

5.16.3 Member Function Documentation

5.16.3.1 String crea.recibePort.getCreararchivo (@QueryParam("nombre") String nombre, @QueryParam("contenido") String contenido, @QueryParam("usuario") String usuario, @QueryParam("password") String password)

Invokes the SOAP method `creararchivo`

Parameters

nombre resource URI parameter
contenido resource URI parameter

Returns

an instance of `java.lang.String`

Definition at line 43 of file [recibePort.java](#).

The documentation for this class was generated from the following file:

- [src/java/crea/recibePort.java](#)

5.17 valor.tamanio Class Reference

Public Member Functions

- String [archivokb](#) (@WebParam(name="nombre")
String nombre, @WebParam(name="usuario") String
usuario, @WebParam(name="password") String password)

5.17.1 Detailed Description

Author

Rodrigo Ramírez Reyes

Definition at line 25 of file [tamanio.java](#).

5.17.2 Member Function Documentation

- 5.17.2.1** String [valor.tamanio.archivokb](#) (@WebParam(name="nombre")
String *nombre*, @WebParam(name="usuario") String *usuario*,
@WebParam(name="password") String *password*)

Web service operation Regresa el tamaño de un archivo del directorio de archivos del servidor.

Parameters

nombre Nombre de un archivo

usuario Nombre de usuario que solicita el servicio

password Password cifrado del usuario que solicita el servicio

Returns

Regresa el tamaño del archivo solicitado

Definition at line 41 of file [tamanio.java](#).

The documentation for this class was generated from the following file:

- [src/java/valor/tamanio.java](#)

5.18 valor.tamanoPort Class Reference

Public Member Functions

- `tamanoPort ()`
- `String getArchivokb (@QueryParam("nombre") String nombre, @QueryParam("usuario") String usuario, @QueryParam("password") String password)`

5.18.1 Detailed Description

REST Web Service

Author

Rodrigo

Definition at line 23 of file `tamanoPort.java`.

5.18.2 Constructor & Destructor Documentation

5.18.2.1 valor.tamanoPort.tamanoPort ()

Creates a new instance of `tamanoPort`

Definition at line 29 of file `tamanoPort.java`.

5.18.3 Member Function Documentation

5.18.3.1 String valor.tamanoPort.getArchivokb (@QueryParam("nombre") String nombre, @QueryParam("usuario") String usuario, @QueryParam("password") String password)

Invokes the SOAP method `archivokb`

Parameters

nombre resource URI parameter

usuario resource URI parameter

password resource URI parameter

Returns

an instance of `java.lang.String`

Definition at line 44 of file [tamanoPort.java](#).

The documentation for this class was generated from the following file:

- [src/java/valor/tamanoPort.java](#)

5.19 `visualiza.todosarchivos` Class Reference

Public Member Functions

- String `lista` (`@WebParam(name="usuario")` String `usuario`, `@WebParam(name="password")` String `password`)

5.19.1 Detailed Description

Author

Rodrigo Ramírez Reyes

Definition at line 24 of file `todosarchivos.java`.

5.19.2 Member Function Documentation

5.19.2.1 String `visualiza.todosarchivos.lista` (`@WebParam(name="usuario")` String `usuario`, `@WebParam(name="password")` String `password`)

Web service operation Regresa la lista completa de los archivos del servidor

Parameters

usuario Nombre de usuario que solicita el servicio

password Password cifrado del usuario que solicita el servicio

Returns

Listado de archivos del servidor

Definition at line 39 of file `todosarchivos.java`.

The documentation for this class was generated from the following file:

- `src/java/visualiza/todosarchivos.java`

5.20 `visualiza.todosarchivosPort` Class Reference

Public Member Functions

- `todosarchivosPort ()`
- `String visualiza.todosarchivosPort.getLista (@QueryParam("usuario") String usuario, @QueryParam("password") String password)` `String`

5.20.1 Detailed Description

REST Web Service

Author

Rodrigo

Definition at line 24 of file `todosarchivosPort.java`.

5.20.2 Constructor & Destructor Documentation

5.20.2.1 `visualiza.todosarchivosPort.todosarchivosPort ()`

Creates a new instance of `todosarchivosPort`

Definition at line 30 of file `todosarchivosPort.java`.

5.20.3 Member Function Documentation

5.20.3.1 `String visualiza.todosarchivosPort.getLista (@QueryParam("usuario") String usuario, @QueryParam("password") String password)`

Invokes the SOAP method `lista`

Parameters

usuario resource URI parameter

Returns

an instance of `java.lang.String`

Definition at line 43 of file `todosarchivosPort.java`.

The documentation for this class was generated from the following file:

- [src/java/visualiza/todosarchivosPort.java](#)

5.21 acceso.user Class Reference

Public Member Functions

- String `valida` (`@WebParam(name="user") String user, @WebParam(name="pass") String pass`) String

5.21.1 Detailed Description

Author

Rodrigo Ramírez Reyes

Definition at line 23 of file `user.java`.

5.21.2 Member Function Documentation

5.21.2.1 String `acceso.user.valida` (`@WebParam(name="user") String user, @WebParam(name="pass") String pass`)

Web service operation Valida el inicio de sesión en la aplicación cliente, mediante la validación de la sesión en el servidor de archivos.

Parameters

user Nombre de usuario que solicita el servicio

pass Password cifrado del usuario que solicita el servicio

Returns

Regresa true si los datos son correctos en el servidor de archivos

Definition at line 38 of file `user.java`.

The documentation for this class was generated from the following file:

- `src/java/acceso/user.java`

5.22 acceso.userPort Class Reference

Public Member Functions

- [userPort \(\)](#)
- [String getValida \(@QueryParam\("user"\) String user,@QueryParam\("pass"\) String pass\)](#)

5.22.1 Detailed Description

REST Web Service

Author

Rodrigo

Definition at line 24 of file [userPort.java](#).

5.22.2 Constructor & Destructor Documentation

5.22.2.1 acceso.userPort.userPort ()

Creates a new instance of [userPort](#)

Definition at line 30 of file [userPort.java](#).

5.22.3 Member Function Documentation

5.22.3.1 String acceso.userPort.getValida (@QueryParam("user") String user, @QueryParam("pass") String pass)

Invokes the SOAP method valida

Parameters

user resource URI parameter

pass resource URI parameter

Returns

an instance of `java.lang.String`

Definition at line 44 of file [userPort.java](#).

The documentation for this class was generated from the following file:

- [src/java/acceso/userPort.java](#)

Chapter 6

File Documentation

6.1 src/java/acceso/user.java File Reference

Classes

- class [acceso.user](#)

Packages

- package [acceso](#)

6.2 src/java/acceso/user.java

```
00001 /*
00002  * To change this template, choose Tools | Templates
00003  * and open the template in the editor.
00004  */
00005
00006 package acceso;
00007
00008 import java.io.IOException;
00009 import java.net.SocketException;
00010 import java.util.logging.Level;
00011 import java.util.logging.Logger;
00012 import javax.jws.WebMethod;
00013 import javax.jws.WebParam;
00014 import javax.jws.WebService;
00015 import org.apache.commons.net.ftp.FTPClient;
00016
00022 @WebService()
00023 public class user {
00024
00037     @WebMethod(operationName = "valida")
00038     public String valida(@WebParam(name = "user")
00039         String user, @WebParam(name = "pass")
00040         String pass) {
00041         //TODO write your implementation code here:
00042
00043         //realizar una comprobación de acceso al servidor ftp
00044
00045         FTPClient f;
00046
00047
00048         try {
00049
00050             f = new FTPClient();
00051             f.connect("127.0.0.1");
00052
00053             if(f.login(user, pass)){
00054                 f.logout();
00055                 return "true";
00056             }
00057
00058         } catch (SocketException ex) {
00059             Logger.getLogger(user.class.getName()).log(Level.SEVERE, null, ex);
00060         } catch (IOException ex) {
00061             Logger.getLogger(user.class.getName()).log(Level.SEVERE, null, ex);
00062         }
00063
00064         return "false";
00065
00066     }
00067
00068 }
```


6.3 src/java/acceso/userPort.java File Reference

Classes

- class [acceso.userPort](#)

Packages

- package [acceso](#)

6.4 src/java/acceso/userPort.java

```
00001 /*
00002  * To change this template, choose Tools | Templates
00003  * and open the template in the editor.
00004  */
00005
00006 package acceso;
00007
00008 import acceso_client.User;
00009 import javax.ws.rs.Consumes;
00010 import javax.ws.rs.GET;
00011 import javax.ws.rs.Path;
00012 import javax.ws.rs.Produces;
00013 import javax.ws.rs.QueryParam;
00014 import javax.ws.rs.core.Context;
00015 import javax.ws.rs.core.UriInfo;
00016
00023 @Path("userport")
00024 public class userPort {
00025     private User port;
00026     @Context
00027     private UriInfo context;
00028
00030     public userPort() {
00031         port = getPort();
00032     }
00033
00040     @GET
00041     @Produces("text/plain")
00042     @Consumes("text/plain")
00043     @Path("valida/")
00044     public String getValida(@QueryParam("user")
00045     String user, @QueryParam("pass")
00046     String pass) {
00047         try {
00048             // Call Web Service Operation
00049             if (port != null) {
00050                 java.lang.String result = port.valida(user, pass);
00051                 return result;
00052             }
00053         } catch (Exception ex) {
00054             // TODO handle custom exceptions here
00055         }
00056         return null;
00057     }
00058
00062     private User getPort() {
00063         try {
00064             // Call Web Service Operation
00065             acceso_client.UserService service = new acceso_client.UserService();
00066             acceso_client.User p = service.getUserPort();
00067             return p;
00068         } catch (Exception ex) {
00069             // TODO handle custom exceptions here
00070         }
00071         return null;
00071     }
00071 }
```

```
00072     }  
00073 }
```

6.5 src/java/archivodate/fecha.java File Reference

Classes

- class [archivodate.fecha](#)

Packages

- package [archivodate](#)

6.6 src/java/archivodate/fecha.java

```
00001 /*
00002  * To change this template, choose Tools | Templates
00003  * and open the template in the editor.
00004  */
00005
00006 package archivodate;
00007
00008 import java.io.IOException;
00009 import java.net.SocketException;
00010 import java.util.logging.Level;
00011 import java.util.logging.Logger;
00012 import javax.jws.WebMethod;
00013 import javax.jws.WebParam;
00014 import javax.jws.WebService;
00015 import org.apache.commons.net.ftp.FTPClient;
00016 import org.apache.commons.net.ftp.FTPFile;
00017 import org.apache.commons.net.ftp.FTPListParseEngine;
00018
00024 @WebService()
00025 public class fecha {
00026
00040     @WebMethod(operationName = "fechaedicion")
00041     public String fechaedicion(@WebParam(name = "nombre")
00042     String nombre, @WebParam(name = "usuario")
00043     String usuario, @WebParam(name = "password")
00044     String password) {
00045         //TODO write your implementation code here:
00046
00047         String file = "";
00048         FTPClient f;
00049         FTPListParseEngine motor;
00050         FTPFile[] archivos;
00051
00052
00053         try {
00054             f = new FTPClient();
00055             f.connect("127.0.0.1");
00056             f.login(usuario, password);
00057
00058             motor = f.initiateListParsing("/");
00059
00060             archivos = motor.getFiles();
00061
00062             for(int i = 0; i < archivos.length; i++){
00063                 if(nombre.equals(archivos[i].getName())){
00064                     f.logout();
00065                     return String.valueOf(archivos[i].getTimestamp().getTime());
00066                 }
00067             }
00068
00069             f.logout();
00070
00071         } catch (SocketException ex) {
00072             Logger.getLogger(fecha.class.getName()).log(Level.SEVERE, null, ex);
00073         } catch (IOException ex) {
```

```
00074         Logger.getLogger(fecha.class.getName()).log(Level.SEVERE, null, ex);
00075     }
00076
00077     return null;
00078 }
00079
00080 }
```


6.7 src/java/archivodate/fechaPort.java File Reference

Classes

- class [archivodate.fechaPort](#)

Packages

- package [archivodate](#)

6.8 src/java/archivodate/fechaPort.java

```

00001  /*
00002  * To change this template, choose Tools | Templates
00003  * and open the template in the editor.
00004  */
00005
00006  package archivodate;
00007
00008  import javax.ws.rs.Consumes;
00009  import javax.ws.rs.GET;
00010  import javax.ws.rs.Path;
00011  import javax.ws.rs.Produces;
00012  import javax.ws.rs.QueryParam;
00013  import javax.ws.rs.core.Context;
00014  import javax.ws.rs.core.UriInfo;
00015
00022  @Path("fechaport")
00023  public class fechaPort {
00024      private archivodate_client.Fecha port;
00025      @Context
00026      private UriInfo context;
00027
00029      public fechaPort() {
00030          port = getPort();
00031      }
00032
00040      @GET
00041      @Produces("text/plain")
00042      @Consumes("text/plain")
00043      @Path("fechaedicion/*")
00044      public String getFechaedicion(@QueryParam("nombre")
00045      String nombre, @QueryParam("usuario")
00046      String usuario, @QueryParam("password")
00047      String password) {
00048          try {
00049              // Call Web Service Operation
00050              if (port != null) {
00051                  java.lang.String result = port.fechaedicion(nombre, usuario, pass
word);
00052                  return result;
00053              }
00054          } catch (Exception ex) {
00055              // TODO handle custom exceptions here
00056          }
00057          return null;
00058      }
00059
00063      private archivodate_client.Fecha getPort() {
00064          try {
00065              // Call Web Service Operation
00066              archivodate_client.FechaService service = new archivodate_client.Fech
aService();
00067              archivodate_client.Fecha p = service.getFechaPort();
00068              return p;
00069          } catch (Exception ex) {
00070              // TODO handle custom exceptions here

```

```
00071     }  
00072     return null;  
00073 }  
00074 }
```

6.9 src/java/compara/bitacoras.java File Reference

Classes

- class [compara.bitacoras](#)

Packages

- package [compara](#)

6.10 src/java/compara/bitacoras.java

```
00001 /*
00002  * To change this template, choose Tools | Templates
00003  * and open the template in the editor.
00004  */
00005
00006 package compara;
00007
00008 import java.io.IOException;
00009 import java.net.SocketException;
00010 import java.util.logging.Level;
00011 import java.util.logging.Logger;
00012 import javax.jws.WebMethod;
00013 import javax.jws.WebParam;
00014 import javax.jws.WebService;
00015 import org.apache.commons.net.ftp.FTPClient;
00016 import org.apache.commons.net.ftp.FTPFile;
00017 import org.apache.commons.net.ftp.FTPListParseEngine;
00018
00024 @WebService()
00025 public class bitacoras {
00026
00041     @WebMethod(operationName = "comparabitacoras")
00042     public String comparabitacoras(@WebParam(name = "arbolArchivos")
00043     String arbolArchivos, @WebParam(name = "usuario")
00044     String usuario, @WebParam(name = "password")
00045     String password) {
00046         //TODO write your implementation code here:
00047
00048         String bitacoraLocal = "";
00049         FTPClient f;
00050         FTPListParseEngine motor;
00051         FTPFile[] archivos;
00052         String resultado = "";
00053         String tmp = null;
00054         int a = 0;
00055         int b = 0;
00056         int max = 0;
00057
00058         try {
00059
00060             f = new FTPClient();
00061             f.connect("127.0.0.1");
00062             if(f.login(usuario, password)){
00063
00064                 //se obtienen los archivos enviados desde el cliente
00065
00066                 String[] token = arbolArchivos.split(">");
00067                 arbolArchivos = "";
00068
00069                 for (int x = 0; x < token.length; x++){
00070                     token[x] += ">";
00071                     bitacoraLocal += token[x];
00072                 }
00073
00074                 //se listan los archivos del servidor
```

```

00075
00076         motor = f.initiateListParsing("/");
00077
00078         archivos = motor.GetFiles();
00079
00080         bitacoraLocal += "\n";
00081
00082         for(int i = 0; i < archivos.length; i++){
00083             if(archivos[i].isDirectory()){
00084                 tmp = archivos[i].getName();
00085                 archivos[i].setName("<" + tmp + ">");
00086                 bitacoraLocal += archivos[i].getName();
00087             }
00088             else if(archivos[i].isFile()){
00089                 tmp = archivos[i].getName();
00090                 archivos[i].setName("<" + tmp + ">");
00091                 bitacoraLocal += archivos[i].getName();
00092             }
00093         }
00094
00095         for(int g = 0; g < archivos.length; g++){
00096             for(int j = 0; j < token.length; j++){
00097                 if(archivos[g].getName().equals(token[j])){
00098                     break;
00099                 }
00100                 else if(j == (token.length-1)){
00101                     resultado += archivos[g].getName() + "\n";
00102                 }
00103             }
00104         }
00105
00106         f.logout();
00107         return resultado;
00108
00109     }
00110
00111 }
00112
00113 } catch (SocketException ex) {
00114     Logger.getLogger(bitacoras.class.getName()).log(Level.SEVERE, null, e
x);
00115 } catch (IOException ex) {
00116     Logger.getLogger(bitacoras.class.getName()).log(Level.SEVERE, null, e
x);
00117 }
00118
00119 return "";
00120
00121 }
00122
00123 }

```


6.11 src/java/compara/bitacorasPort.java File Reference

Classes

- class [compara.bitacorasPort](#)

Packages

- package [compara](#)

6.12 src/java/compara/bitacorasPort.java

```

00001 /*
00002  * To change this template, choose Tools | Templates
00003  * and open the template in the editor.
00004  */
00005
00006 package compara;
00007
00008 import javax.ws.rs.Consumes;
00009 import javax.ws.rs.GET;
00010 import javax.ws.rs.Path;
00011 import javax.ws.rs.Produces;
00012 import javax.ws.rs.QueryParam;
00013 import javax.ws.rs.core.Context;
00014 import javax.ws.rs.core.UriInfo;
00015
00022 @Path("bitacorasport")
00023 public class bitacorasPort {
00024     private compara_client.Bitacoras port;
00025     @Context
00026     private UriInfo context;
00027
00029     public bitacorasPort() {
00030         port = getPort();
00031     }
00032
00038     @GET
00039     @Produces("text/plain")
00040     @Consumes("text/plain")
00041     @Path("comparabitacoras/")
00042     public String getComparabitacoras(@QueryParam("arbolArchivos")
00043     String arbolArchivos, @QueryParam("usuario")
00044     String usuario, @QueryParam("password")
00045     String password) {
00046         try {
00047             // Call Web Service Operation
00048             if (port != null) {
00049                 java.lang.String result = port.comparabitacoras(arbolArchivos, us
00050                 uario, password);
00051                 return result;
00052             } catch (Exception ex) {
00053                 // TODO handle custom exceptions here
00054             }
00055             return null;
00056         }
00057
00061     private compara_client.Bitacoras getPort() {
00062         try {
00063             // Call Web Service Operation
00064             compara_client.BitacorasService service = new compara_client.Bitacora
00065             sService();
00066             compara_client.Bitacoras p = service.getBitacorasPort();
00067             return p;
00068         } catch (Exception ex) {
00069             // TODO handle custom exceptions here

```

```
00069     }  
00070     return null;  
00071 }  
00072 }
```

6.13 src/java/comparaup/comparabitacoraup.java File Reference

Classes

- class [comparaup.comparabitacoraup](#)

Packages

- package [comparaup](#)

6.14 src/java/comparaup/comparabitacoraup.java

```
00001 /*
00002  * To change this template, choose Tools | Templates
00003  * and open the template in the editor.
00004  */
00005
00006 package comparaup;
00007
00008 import java.io.IOException;
00009 import java.net.SocketException;
00010 import java.util.logging.Level;
00011 import java.util.logging.Logger;
00012 import javax.jws.WebMethod;
00013 import javax.jws.WebParam;
00014 import javax.jws.WebService;
00015 import org.apache.commons.net.ftp.FTPClient;
00016 import org.apache.commons.net.ftp.FTPFile;
00017 import org.apache.commons.net.ftp.FTPListParseEngine;
00018
00024 @WebService()
00025 public class comparabitacoraup {
00026
00041     @WebMethod(operationName = "comparabitacoras")
00042     public String comparabitacoras(@WebParam(name = "arbolArchivos")
00043     String arbolArchivos, @WebParam(name = "usuario")
00044     String usuario, @WebParam(name = "password")
00045     String password) {
00046         //TODO write your implementation code here:
00047
00048         String bitacoraLocal = "";
00049         FTPClient f;
00050         FTPListParseEngine motor;
00051         FTPFile[] archivos;
00052         String resultado = "";
00053         String tmp = null;
00054         int a = 0;
00055         int b = 0;
00056         int max = 0;
00057
00058         try {
00059
00060             f = new FTPClient();
00061             f.connect("127.0.0.1");
00062             if(f.login(usuario, password)){
00063
00064                 //se obtienen los archivos enviados desde el cliente
00065
00066                 String[] token = arbolArchivos.split(">");
00067                 arbolArchivos = "";
00068
00069                 for (int x = 0; x < token.length; x++){
00070                     token[x] += ">";
00071                     bitacoraLocal += token[x];
00072                 }
00073
00074                 //se listan los archivos del servidor
```

```
00075
00076         motor = f.initiateListParsing("/");
00077
00078         archivos = motor.GetFiles();
00079
00080         bitacoraLocal += "\n";
00081
00082         for(int i = 0; i < archivos.length; i++){
00083             if(archivos[i].isDirectory()){
00084                 tmp = archivos[i].getName();
00085                 archivos[i].setName("<" + tmp + ">");
00086                 bitacoraLocal += archivos[i].getName();
00087             }
00088             else if(archivos[i].isFile()){
00089                 tmp = archivos[i].getName();
00090                 archivos[i].setName("<" + tmp + ">");
00091                 bitacoraLocal += archivos[i].getName();
00092             }
00093         }
00094
00095         for(int g = 0; g < token.length; g++){
00096             for(int j = 0; j < archivos.length; j++){
00097                 if(token[g].equals(archivos[j].getName())){
00098                     break;
00099                 }
00100                 else if(j == (archivos.length-1)){
00101                     resultado += token[g] + "\n";
00102                 }
00103             }
00104         }
00105
00106         f.logout();
00107         return resultado;
00108     }
00109 }
00110
00111 } catch (SocketException ex) {
00112     Logger.getLogger(comparabitacoraup.class.getName()).log(Level.SEVERE,
00113 null, ex);
00114 } catch (IOException ex) {
00115     Logger.getLogger(comparabitacoraup.class.getName()).log(Level.SEVERE,
00116 null, ex);
00117 }
00118 return "";
00119 }
00120 }
00121 }
00122 }
```


6.15 src/java/comparaup/comparabitacoraupPort.java File Reference

Classes

- class [comparaup.comparabitacoraupPort](#)

Packages

- package [comparaup](#)

6.16 src/java/comparaup/comparabitacoraupPort.java

```

00001 /*
00002  * To change this template, choose Tools | Templates
00003  * and open the template in the editor.
00004  */
00005
00006 package comparaup;
00007
00008 import javax.ws.rs.Consumes;
00009 import javax.ws.rs.GET;
00010 import javax.ws.rs.Path;
00011 import javax.ws.rs.Produces;
00012 import javax.ws.rs.QueryParam;
00013 import javax.ws.rs.core.Context;
00014 import javax.ws.rs.core.UriInfo;
00015
00022 @Path("comparabitacoraupport")
00023 public class comparabitacoraupPort {
00024     private comparaup_client.Comparabitacoraup port;
00025     @Context
00026     private UriInfo context;
00027
00029     public comparabitacoraupPort() {
00030         port = getPort();
00031     }
00032
00038     @GET
00039     @Produces("text/plain")
00040     @Consumes("text/plain")
00041     @Path("comparabitacoras/")
00042     public String getComparabitacoras(@QueryParam("arbolArchivos")
00043     String arbolArchivos, @QueryParam("usuario")
00044     String usuario, @QueryParam("password")
00045     String password) {
00046         try {
00047             // Call Web Service Operation
00048             if (port != null) {
00049                 java.lang.String result = port.comparabitacoras(arbolArchivos, us
00050                 uario, password);
00051                 return result;
00052             } catch (Exception ex) {
00053                 // TODO handle custom exceptions here
00054             }
00055             return null;
00056         }
00057
00061     private comparaup_client.Comparabitacoraup getPort() {
00062         try {
00063             // Call Web Service Operation
00064             comparaup_client.ComparabitacoraupService service = new comparaup_cli
00065             ent.ComparabitacoraupService();
00066             comparaup_client.Comparabitacoraup p = service.getComparabitacoraupPo
00067             rt();
00068             return p;
00069         } catch (Exception ex) {

```

```
00068         // TODO handle custom exceptions here
00069     }
00070     return null;
00071 }
00072 }
```

6.17 src/java/crea/recibe.java File Reference

Classes

- class [crea.recibe](#)

Packages

- package [crea](#)

6.18 src/java/crea/recibe.java

```
00001 /*
00002  * To change this template, choose Tools | Templates
00003  * and open the template in the editor.
00004  */
00005
00006 package crea;
00007
00008 import java.io.ByteArrayInputStream;
00009 import java.io.IOException;
00010 import java.io.InputStream;
00011 import java.io.UnsupportedEncodingException;
00012 import java.net.SocketException;
00013 import java.util.StringTokenizer;
00014 import java.util.logging.Level;
00015 import java.util.logging.Logger;
00016 import javax.jws.WebMethod;
00017 import javax.jws.WebParam;
00018 import javax.jws.WebService;
00019 import org.apache.commons.net.ftp.FTPClient;
00020 import org.apache.commons.net.ftp.FTPFile;
00021 import org.apache.commons.net.ftp.FTPListParseEngine;
00022
00023
00029 @WebService()
00030 public class recibe {
00031
00047     @WebMethod(operationName = "creararchivo")
00048     public String creararchivo(@WebParam(name = "nombre")
00049     String nombre, @WebParam(name = "contenido")
00050     String contenido, @WebParam(name = "usuario")
00051     String usuario, @WebParam(name = "password")
00052     String password) {
00053
00054         try {
00055             //TODO write your implementation code here:
00056
00057             //conectamos hacia el servidor
00058             FTPClient f;
00059             f = new FTPClient();
00060             f.connect("127.0.0.1");
00061             f.login(usuario, password);
00062             String test = "";
00063             String bueno = "";
00064
00065             //cambiar el string a un inputstream
00066             InputStream is = new ByteArrayInputStream(contenido.getBytes("UTF-8"))
00067 );
00068             //InputStream is = new ByteArrayInputStream(contenido.getBytes());
00069             //colocamos el directorio en la raíz
00070             f.setFileType(FTPClient.BINARY_FILE_TYPE);
00071             if(nombre.startsWith("/")){
00072                 String[] token = nombre.split("/");
00073                 for (int x = 0; x < token.length; x++){
00074                     if(x > 0){
00075                         if(x == (token.length - 1))
```

```
00075         bueno += token[x];
00076     else
00077         test += token[x] + "/";
00078     }
00079     else
00080         test += "/";
00081     }
00082
00083     f.makeDirectory(test);
00084     f.changeWorkingDirectory(test);
00085     //subimos el archivo
00086     if(f.storeFile(bueno, is)){
00087         f.logout();
00088         return "true";
00089     }
00090
00091
00092     }
00093     else{
00094         //subimos el archivo
00095         if(f.storeFile(nombre, is)){
00096             f.logout();
00097             return "true";
00098         }
00099     }
00100
00101     }
00102
00103     } catch (SocketException ex) {
00104         Logger.getLogger(recibe.class.getName()).log(Level.SEVERE, null, ex);
00105
00106     } catch (IOException ex) {
00107         Logger.getLogger(recibe.class.getName()).log(Level.SEVERE, null, ex);
00108     }
00109     return "false";
00110 }
00111
00112
00113
00114 }
```


6.19 src/java/crea/recibePort.java File Reference

Classes

- class [crea.recibePort](#)

Packages

- package [crea](#)

6.20 src/java/crea/recibePort.java

```

00001 /*
00002  * To change this template, choose Tools | Templates
00003  * and open the template in the editor.
00004  */
00005
00006 package crea;
00007
00008 import javax.ws.rs.Consumes;
00009 import javax.ws.rs.GET;
00010 import javax.ws.rs.Path;
00011 import javax.ws.rs.Produces;
00012 import javax.ws.rs.QueryParam;
00013 import javax.ws.rs.core.Context;
00014 import javax.ws.rs.core.UriInfo;
00015
00022 @Path("/recibeport")
00023 public class recibePort {
00024     private crea_client.Recibe port;
00025     @Context
00026     private UriInfo context;
00027
00029     public recibePort() {
00030         port = getPort();
00031     }
00032
00039     @GET
00040     @Produces("text/plain")
00041     @Consumes("text/plain")
00042     @Path("/creararchivo/*")
00043     public String getCreararchivo(@QueryParam("nombre")
00044     String nombre, @QueryParam("contenido")
00045     String contenido, @QueryParam("usuario")
00046     String usuario, @QueryParam("password")
00047     String password) {
00048         try {
00049             // Call Web Service Operation
00050             if (port != null) {
00051                 java.lang.String result = port.creararchivo(nombre, contenido, us
00052                 uario, password);
00053                 return result;
00054             } catch (Exception ex) {
00055                 // TODO handle custom exceptions here
00056             }
00057             return null;
00058         }
00059
00063     private crea_client.Recibe getPort() {
00064         try {
00065             // Call Web Service Operation
00066             crea_client.RecibeService service = new crea_client.RecibeService();
00067             crea_client.Recibe p = service.getRecibePort();
00068             return p;
00069         } catch (Exception ex) {
00070             // TODO handle custom exceptions here

```

```
00071     }  
00072     return null;  
00073 }  
00074 }
```

6.21 src/java/obtienearchivo/archivo.java File Reference

Classes

- class [obtienearchivo.archivo](#)

Packages

- package [obtienearchivo](#)

6.22 src/java/obtienearchivo/archivo.java

```
00001 /*
00002  * To change this template, choose Tools | Templates
00003  * and open the template in the editor.
00004  */
00005
00006 package obtienearchivo;
00007
00008 import java.io.BufferedReader;
00009 import java.io.IOException;
00010 import java.io.InputStream;
00011 import java.io.InputStreamReader;
00012 import java.net.SocketException;
00013 import java.util.logging.Level;
00014 import java.util.logging.Logger;
00015 import javax.jws.WebMethod;
00016 import javax.jws.WebParam;
00017 import javax.jws.WebService;
00018 import org.apache.commons.net.ftp.FTP;
00019 import org.apache.commons.net.ftp.FTPClient;
00020 import org.apache.commons.net.ftp.FTPFile;
00021 import org.apache.commons.net.ftp.FTPListParseEngine;
00022
00023 @WebService()
00024 public class archivo {
00025
00026     @WebMethod(operationName = "descarga")
00027     public String descarga(@WebParam(name = "nombreamonivo")
00028         String nombreamonivo, @WebParam(name = "usuario")
00029         String usuario, @WebParam(name = "password")
00030         String password) {
00031         //TODO write your implementation code here:
00032
00033         String file = "";
00034         FTPClient f;
00035         FTPListParseEngine motor;
00036         FTPFile[] archivos;
00037         InputStream is;
00038         StringBuilder sb = new StringBuilder();
00039         String line;
00040
00041         try {
00042
00043             f = new FTPClient();
00044             f.connect("127.0.0.1");
00045             f.login(usuario, password);
00046
00047             is = f.retrieveFileStream(nombreamonivo);
00048             String salida = is.toString();
00049             byte[] out = salida.getBytes();
00050
00051             f.logout();
00052             return out.toString();
00053
00054         }
00055     }
00056 }
00057
00058
00059
00060
00061
00062
00063
00064
00065
00066
00067
00068
00069
00070
00071
00072
```

```
00073     } catch (SocketException ex) {
00074         Logger.getLogger(archivo.class.getName()).log(Level.SEVERE, null, ex)
00075     ;
00076     } catch (IOException ex) {
00077         Logger.getLogger(archivo.class.getName()).log(Level.SEVERE, null, ex)
00078     ;
00079     }
00080     }
00081     return null;
00082 }
00083 }
```


6.23 src/java/obtienearchivo/archivoPort.java File Reference

Classes

- class [obtienearchivo.archivoPort](#)

Packages

- package [obtienearchivo](#)

6.24 src/java/obtienearchivo/archivoPort.java

```

00001  /*
00002  * To change this template, choose Tools | Templates
00003  * and open the template in the editor.
00004  */
00005
00006  package obtienearchivo;
00007
00008  import javax.ws.rs.Consumes;
00009  import javax.ws.rs.GET;
00010  import javax.ws.rs.Path;
00011  import javax.ws.rs.Produces;
00012  import javax.ws.rs.QueryParam;
00013  import javax.ws.rs.core.Context;
00014  import javax.ws.rs.core.UriInfo;
00015
00022  @Path("archivoport")
00023  public class archivoPort {
00024      private obtienearchivo_client.Archivo port;
00025      @Context
00026      private UriInfo context;
00027
00029      public archivoPort () {
00030          port = getPort ();
00031      }
00032
00038      @GET
00039      @Produces("text/plain")
00040      @Consumes("text/plain")
00041      @Path("descarga/")
00042      public String getDescarga(@QueryParam("nombreachivo")
00043      String nombreachivo, @QueryParam("usuario")
00044      String usuario, @QueryParam("password")
00045      String password) {
00046          try {
00047              // Call Web Service Operation
00048              if (port != null) {
00049                  java.lang.String result = port.descarga(nombreachivo, usuario, p
00050                  assword);
00051                  return result;
00052              } catch (Exception ex) {
00053                  // TODO handle custom exceptions here
00054              }
00055              return null;
00056          }
00057
00061      private obtienearchivo_client.Archivo getPort () {
00062          try {
00063              // Call Web Service Operation
00064              obtienearchivo_client.ArchivoService service = new obtienearchivo_cli
00065              ent.ArchivoService();
00066              obtienearchivo_client.Archivo p = service.getArchivoPort ();
00067              return p;
00068          } catch (Exception ex) {
00069              // TODO handle custom exceptions here

```

```
00069     }  
00070     return null;  
00071 }  
00072 }
```

6.25 `src/java/reemplazo/archivosreemplazo.java` File Reference

Classes

- class [reemplazo.archivosreemplazo](#)

Packages

- package [reemplazo](#)

6.26 src/java/reemplazo/archivosreemplazo.java

```
00001 /*
00002  * To change this template, choose Tools | Templates
00003  * and open the template in the editor.
00004  */
00005
00006 package reemplazo;
00007
00008 import java.io.IOException;
00009 import java.net.SocketException;
00010 import java.util.logging.Level;
00011 import java.util.logging.Logger;
00012 import javax.jws.WebMethod;
00013 import javax.jws.WebParam;
00014 import javax.jws.WebService;
00015 import org.apache.commons.net.ftp.*;
00016
00022 @WebService()
00023 public class archivosreemplazo {
00024
00037     @WebMethod(operationName = "lista")
00038     public String lista(@WebParam(name = "usuario")
00039         String usuario, @WebParam(name = "password")
00040         String password) {
00041         //TODO write your implementation code here:
00042
00043         String file = "";
00044         FTPClient f;
00045         FTPListParseEngine motor;
00046         FTPFile[] archivos;
00047
00048
00049         try {
00050             f = new FTPClient();
00051             f.connect("127.0.0.1");
00052             f.login(usuario, password);
00053
00054             motor = f.initiateListParsing("/");
00055
00056             archivos = motor.getFiles();
00057
00058             for(int i = 0; i < archivos.length; i++){
00059                 if(archivos[i].isDirectory()){
00060                     file += "<" + archivos[i].getName() + ">\n";
00061                 }
00062                 else if(archivos[i].isFile())
00063                     file += "<" + archivos[i].getName() + ">\n";
00064             }
00065
00066             f.logout();
00067         } catch (SocketException ex) {
00068             Logger.getLogger(archivosreemplazo.class.getName()).log(Level.SEVERE,
00069                 null, ex);
00069         } catch (IOException ex) {
00070             Logger.getLogger(archivosreemplazo.class.getName()).log(Level.SEVERE,
00070                 null, ex);

```

```
00071     }
00072
00073
00074
00075
00076
00077 //     if(usuario.equals("rodrigo")){
00078 //         archivos += "<Documentos/>\n" +
00079 //             "<Descargas/>\n" +
00080 //             "<Imágenes/>\n" +
00081 //             "<Respaldo/>\n" +
00082 //             "<Pendientes/>\n" +
00083 //             "<notas.txt>\n" +
00084 //             "<tareas.txt>\n" +
00085 //             "<cartas.txt>\n" +
00086 //             "<escrito.doc>\n" +
00087 //             "<ambiente.png>\n";
00088 //     }
00089
00090
00091     return file;
00092 }
00093
00094 }
```


6.27 src/java/reemplazo/archivosreemplazoPort.java File Reference

Classes

- class [reemplazo.archivosreemplazoPort](#)

Packages

- package [reemplazo](#)

6.28 src/java/reemplazo/archivosreemplazoPort.java

```

00001 /*
00002  * To change this template, choose Tools | Templates
00003  * and open the template in the editor.
00004  */
00005
00006 package reemplazo;
00007
00008 import javax.ws.rs.Consumes;
00009 import javax.ws.rs.GET;
00010 import javax.ws.rs.Path;
00011 import javax.ws.rs.Produces;
00012 import javax.ws.rs.QueryParam;
00013 import javax.ws.rs.core.Context;
00014 import javax.ws.rs.core.UriInfo;
00015 import reemplazo_client.Archivosreemplazo;
00016
00023 @Path("archivosreemplazoport")
00024 public class archivosreemplazoPort {
00025     private Archivosreemplazo port;
00026     @Context
00027     private UriInfo context;
00028
00030     public archivosreemplazoPort() {
00031         port = getPort();
00032     }
00033
00039     @GET
00040     @Produces("text/plain")
00041     @Consumes("text/plain")
00042     @Path("lista/")
00043     public String getLista(@QueryParam("usuario")
00044     String usuario, @QueryParam("password")
00045     String password) {
00046         try {
00047             // Call Web Service Operation
00048             if (port != null) {
00049                 java.lang.String result = port.lista(usuario, password);
00050                 return result;
00051             }
00052         } catch (Exception ex) {
00053             // TODO handle custom exceptions here
00054         }
00055         return null;
00056     }
00057
00061     private Archivosreemplazo getPort() {
00062         try {
00063             // Call Web Service Operation
00064             reemplazo_client.ArchivosreemplazoService service = new reemplazo_client.ArchivosreemplazoService();
00065             reemplazo_client.Archivosreemplazo p = service.getArchivosreemplazoPort();
00066             return p;
00067         } catch (Exception ex) {
00068             // TODO handle custom exceptions here

```

```
00069     }  
00070     return null;  
00071 }  
00072 }
```

6.29 src/java/regresa/espejo.java File Reference

Classes

- class [regresa.espejo](#)

Packages

- package [regresa](#)

6.30 src/java/regresa/espejo.java

```
00001 /*
00002  * To change this template, choose Tools | Templates
00003  * and open the template in the editor.
00004  */
00005
00006 package regreso;
00007
00008 import javax.ws.WebMethod;
00009 import javax.ws.WebParam;
00010 import javax.ws.WebService;
00011
00012 @WebService()
00013 public class espejo {
00014
00015     @WebMethod(operationName = "igual")
00016     public String igual(@WebParam(name = "arbolarchivos")
00017         String arbolarchivos) {
00018         //TODO write your implementation code here:
00019         String[] token = arbolarchivos.split(">");
00020         arbolarchivos = "";
00021
00022         for (int x = 0; x < token.length; x++){
00023             arbolarchivos += token[x] + ">\n";
00024         }
00025
00026         //<root1/><AccuWeather_AIR.png><acelera.png><archivos.txt>
00027         return arbolarchivos;
00028     }
00029 }
00030 }
```

6.31 src/java/regresa/espejoPort.java File Reference

Classes

- class [regresa.espejoPort](#)

Packages

- package [regresa](#)

6.32 src/java/regresa/espejoPort.java

```
00001 /*
00002  * To change this template, choose Tools | Templates
00003  * and open the template in the editor.
00004  */
00005
00006 package regresa;
00007
00008 import javax.ws.rs.Consumes;
00009 import javax.ws.rs.GET;
00010 import javax.ws.rs.Path;
00011 import javax.ws.rs.Produces;
00012 import javax.ws.rs.QueryParam;
00013 import javax.ws.rs.core.Context;
00014 import javax.ws.rs.core.UriInfo;
00015
00022 @Path("espejoport")
00023 public class espejoPort {
00024     private regresa_client.Espejo port;
00025     @Context
00026     private UriInfo context;
00027
00029     public espejoPort() {
00030         port = getPort();
00031     }
00032
00038     @GET
00039     @Produces("text/plain")
00040     @Consumes("text/plain")
00041     @Path("igual/")
00042     public String getIgual(@QueryParam("arbolarchivos")
00043 String arbolarchivos) {
00044         try {
00045             // Call Web Service Operation
00046             if (port != null) {
00047                 java.lang.String result = port.igual(arbolarchivos);
00048                 return result;
00049             }
00050         } catch (Exception ex) {
00051             // TODO handle custom exceptions here
00052         }
00053         return null;
00054     }
00055
00059     private regresa_client.Espejo getPort() {
00060         try {
00061             // Call Web Service Operation
00062             regresa_client.EspejoService service = new regresa_client.EspejoServi
00063 ce();
00064             regresa_client.Espejo p = service.getEspejoPort();
00065             return p;
00066         } catch (Exception ex) {
00067             // TODO handle custom exceptions here
00068         }
00069         return null;
00070     }
00071 }
```


00070 }

6.33 src/java/valor/tamano.java File Reference

Classes

- class [valor.tamano](#)

Packages

- package [valor](#)

6.34 src/java/valor/tamano.java

```

00001 /*
00002  * To change this template, choose Tools | Templates
00003  * and open the template in the editor.
00004  */
00005
00006 package valor;
00007
00008 import java.io.IOException;
00009 import java.net.SocketException;
00010 import java.util.logging.Level;
00011 import java.util.logging.Logger;
00012 import javax.jws.WebMethod;
00013 import javax.jws.WebParam;
00014 import javax.jws.WebService;
00015 import org.apache.commons.net.ftp.FTPClient;
00016 import org.apache.commons.net.ftp.FTPFile;
00017 import org.apache.commons.net.ftp.FTPListParseEngine;
00018
00024 @WebService()
00025 public class tamano {
00026
00040     @WebMethod(operationName = "archivokb")
00041     public String archivokb(@WebParam(name = "nombre")
00042         String nombre, @WebParam(name="usuario")String usuario,
00043         @WebParam(name="password")String password) {
00044         //TODO write your implementation code here:
00045
00046         String file = "";
00047         FTPClient f;
00048         FTPListParseEngine motor;
00049         FTPFile[] archivos;
00050
00051
00052         try {
00053             f = new FTPClient();
00054             f.connect("127.0.0.1");
00055             f.login(usuario, password);
00056
00057             motor = f.initiateListParsing("/");
00058
00059             archivos = motor.getFiles();
00060
00061             for(int i = 0; i < archivos.length; i++){
00062                 if(nombre.equals(archivos[i].getName())){
00063                     f.logout();
00064                     return String.valueOf(archivos[i].getSize());
00065                 }
00066             }
00067
00068             f.logout();
00069
00070         } catch (SocketException ex) {
00071             Logger.getLogger(tamano.class.getName()).log(Level.SEVERE, null, ex)
00072         };
00072     } catch (IOException ex) {

```

```
00073         Logger.getLogger(tamano.class.getName()).log(Level.SEVERE, null, ex)
00074     };
00075     }
00076
00077
00078     return "0";
00079 }
00080
00081 }
```

6.35 src/java/valor/tamanoPort.java File Reference

Classes

- class [valor.tamanoPort](#)

Packages

- package [valor](#)

6.36 src/java/valor/tamanoPort.java

```
00001 /*
00002  * To change this template, choose Tools | Templates
00003  * and open the template in the editor.
00004  */
00005
00006 package valor;
00007
00008 import javax.ws.rs.Consumes;
00009 import javax.ws.rs.GET;
00010 import javax.ws.rs.Path;
00011 import javax.ws.rs.Produces;
00012 import javax.ws.rs.QueryParam;
00013 import javax.ws.rs.core.Context;
00014 import javax.ws.rs.core.UriInfo;
00015
00022 @Path("tamanoport")
00023 public class tamanoPort {
00024     private valor_client.Tamano port;
00025     @Context
00026     private UriInfo context;
00027
00029     public tamanoPort() {
00030         port = getPort();
00031     }
00032
00040     @GET
00041     @Produces("text/plain")
00042     @Consumes("text/plain")
00043     @Path("archivokb/")
00044     public String getArchivokb(@QueryParam("nombre")
00045     String nombre, @QueryParam("usuario")
00046     String usuario, @QueryParam("password")
00047     String password) {
00048         try {
00049             // Call Web Service Operation
00050             if (port != null) {
00051                 java.lang.String result = port.archivokb(nombre, usuario, passwor
00052 d);
00053                 return result;
00054             }
00054         } catch (Exception ex) {
00055             // TODO handle custom exceptions here
00056         }
00057         return null;
00058     }
00059
00063     private valor_client.Tamano getPort() {
00064         try {
00065             // Call Web Service Operation
00066             valor_client.TamanoService service = new valor_client.TamanoService
00067 ();
00067             valor_client.Tamano p = service.getTamanoPort();
00068             return p;
00069         } catch (Exception ex) {
00070             // TODO handle custom exceptions here
```

```
00071     }  
00072     return null;  
00073 }  
00074 }
```


6.37 src/java/verifica/existe.java File Reference

Classes

- class [verifica.existe](#)

Packages

- package [verifica](#)

6.38 src/java/verifica/existe.java

```
00001 /*
00002  * To change this template, choose Tools | Templates
00003  * and open the template in the editor.
00004  */
00005
00006 package verifica;
00007
00008 import java.io.IOException;
00009 import java.net.SocketException;
00010 import java.util.logging.Level;
00011 import java.util.logging.Logger;
00012 import javax.jws.WebMethod;
00013 import javax.jws.WebParam;
00014 import javax.jws.WebService;
00015 import org.apache.commons.net.ftp.*;
00016
00017
00023 @WebService()
00024 public class existe {
00025
00039     @WebMethod(operationName = "verificaarchivo")
00040     public String verificaarchivo(@WebParam(name = "nombre")
00041     String nombre, @WebParam(name="usuario")String usuario,
00042     @WebParam(name="password")String password) {
00043         //TODO write your implementation code here:
00044
00045         String file = "";
00046         FTPClient f;
00047         FTPListParseEngine motor;
00048         FTPFile[] archivos;
00049
00050         try {
00051             f = new FTPClient();
00052             f.connect("127.0.0.1");
00053             f.login(usuario, password);
00054
00055             motor = f.initiateListParsing("/");
00056
00057             archivos = motor.listFiles();
00058
00059             for(int i = 0; i < archivos.length; i++){
00060                 if(nombre.equals(archivos[i].getName())){
00061                     f.logout();
00062                     return "true";
00063                 }
00064             }
00065
00066             f.logout();
00067
00068         } catch (SocketException ex) {
00069             Logger.getLogger(existe.class.getName()).log(Level.SEVERE, null, ex);
00070
00071         } catch (IOException ex) {
00072             Logger.getLogger(existe.class.getName()).log(Level.SEVERE, null, ex);
00073         }
00074     }
00075 }
```

```
00072     }
00073
00074     return "false";
00075 }
00076
00077 }
```

6.39 src/java/verifica/existePort.java File Reference

Classes

- class [verifica.existePort](#)

Packages

- package [verifica](#)

6.40 src/java/verifica/existePort.java

```

00001 /*
00002  * To change this template, choose Tools | Templates
00003  * and open the template in the editor.
00004  */
00005
00006 package verifica;
00007
00008 import javax.ws.rs.Consumes;
00009 import javax.ws.rs.GET;
00010 import javax.ws.rs.Path;
00011 import javax.ws.rs.Produces;
00012 import javax.ws.rs.QueryParam;
00013 import javax.ws.rs.core.Context;
00014 import javax.ws.rs.core.UriInfo;
00015 import verifica_client.Existe;
00016
00023 @Path("existeport")
00024 public class existePort {
00025     private Existe port;
00026     @Context
00027     private UriInfo context;
00028
00030     public existePort() {
00031         port = getPort();
00032     }
00033
00040     @GET
00041     @Produces("text/plain")
00042     @Consumes("text/plain")
00043     @Path("verificaarchivo/")
00044     public String getVerificaarchivo(@QueryParam("nombre")
00045     String nombre,@QueryParam("usuario")String usuario,
00046     @QueryParam("password")String password) {
00047
00048         try {
00049             // Call Web Service Operation
00050             if (port != null) {
00051                 java.lang.String result = port.verificaarchivo(nombre,usuario,pa
00052                 sword);
00053                 return result;
00054             } catch (Exception ex) {
00055                 // TODO handle custom exceptions here
00056             }
00057             return null;
00058         }
00059
00063     private Existe getPort() {
00064         try {
00065             // Call Web Service Operation
00066             verifica_client.ExisteService service = new verifica_client.ExisteSer
00067             vice();
00068             verifica_client.Existe p = service.getExistePort();
00069             return p;
00069         } catch (Exception ex) {

```

```
00070         // TODO handle custom exceptions here
00071     }
00072     return null;
00073 }
00074 }
```

6.41 src/java/visualiza/todosarchivos.java File Reference

Classes

- class [visualiza.todosarchivos](#)

Packages

- package [visualiza](#)

6.42 src/java/visualiza/todosarchivos.java

```
00001 /*
00002  * To change this template, choose Tools | Templates
00003  * and open the template in the editor.
00004  */
00005
00006 package visualiza;
00007
00008 import java.io.IOException;
00009 import java.util.logging.Level;
00010 import java.util.logging.Logger;
00011 import javax.jws.WebMethod;
00012 import javax.jws.WebParam;
00013 import javax.jws.WebService;
00014 import org.apache.commons.net.ftp.FTPClient;
00015 import org.apache.commons.net.ftp.FTPFile;
00016 import org.apache.commons.net.ftp.FTPListParseEngine;
00017
00023 @WebService()
00024 public class todosarchivos {
00025
00038     @WebMethod(operationName = "lista")
00039     public String lista(@WebParam(name = "usuario")
00040         String usuario, @WebParam(name = "password")
00041         String password) {
00042         //TODO write your implementation code here:
00043
00044         String file = "";
00045         FTPClient f;
00046         FTPListParseEngine motor;
00047         FTPFile[] archivos;
00048
00049         try {
00050             f = new FTPClient();
00051             f.connect("127.0.0.1");
00052             f.login(usuario, password);
00053
00054             motor = f.initiateListParsing("/");
00055
00056             archivos = motor.listFiles();
00057
00058             for(int i = 0; i < archivos.length; i++){
00059                 if(archivos[i].isDirectory()){
00060                     file += "<" + archivos[i].getName() + ">\n";
00061                 }
00062                 else if(archivos[i].isFile())
00063                     file += "<" + archivos[i].getName() + ">\n";
00064             }
00065
00066             f.logout();
00067
00068         } catch (IOException ex) {
00069             Logger.getLogger(todosarchivos.class.getName()).log(Level.SEVERE, nul
00070             l, ex);
00071         }
00071     }
}
```

```
00072
00073
00074
00075     return file;
00076 }
00077
00078 }
```

6.43 src/java/visualiza/todosarchivosPort.java File Reference

Classes

- class [visualiza.todosarchivosPort](#)

Packages

- package [visualiza](#)

6.44 src/java/visualiza/todosarchivosPort.java

```
00001 /*
00002  * To change this template, choose Tools | Templates
00003  * and open the template in the editor.
00004  */
00005
00006 package visualiza;
00007
00008 import javax.ws.rs.Consumes;
00009 import javax.ws.rs.GET;
00010 import javax.ws.rs.Path;
00011 import javax.ws.rs.Produces;
00012 import javax.ws.rs.QueryParam;
00013 import javax.ws.rs.core.Context;
00014 import javax.ws.rs.core.UriInfo;
00015 import visualiza_client.Todosarchivos;
00016
00023 @Path("todosarchivosport")
00024 public class todosarchivosPort {
00025     private Todosarchivos port;
00026     @Context
00027     private UriInfo context;
00028
00030     public todosarchivosPort() {
00031         port = getPort();
00032     }
00033
00039     @GET
00040     @Produces("text/plain")
00041     @Consumes("text/plain")
00042     @Path("lista/")
00043     public String getLista(@QueryParam("usuario")
00044     String usuario, @QueryParam("password")
00045     String password) {
00046         try {
00047             // Call Web Service Operation
00048             if (port != null) {
00049                 java.lang.String result = port.lista(usuario, password);
00050                 return result;
00051             }
00052         } catch (Exception ex) {
00053             // TODO handle custom exceptions here
00054         }
00055         return null;
00056     }
00057
00061     private Todosarchivos getPort() {
00062         try {
00063             // Call Web Service Operation
00064             visualiza_client.TodosarchivosService service = new visualiza_client.
TodosarchivosService();
00065             visualiza_client.Todosarchivos p = service.getTodosarchivosPort();
00066             return p;
00067         } catch (Exception ex) {
00068             // TODO handle custom exceptions here
00069         }
00070     }
00071 }
```

```
00070     return null;
00071   }
00072 }
```

Index

- acceso, 7
- acceso::user, 49
 - valida, 49
- acceso::userPort, 50
 - getValida, 50
 - userPort, 50
- archivodate, 8
- archivodate::fecha, 37
 - fechaedicion, 37
- archivodate::fechaPort, 38
 - fechaPort, 38
 - getFechaedicion, 38
- archivokb
 - valor::tamano, 43
- archivoPort
 - obtienearchivo::archivoPort, 21
- archivosreemplazoPort
 - reemplazo::archivosreemplazoPort, 24

- bitacorasPort
 - compara::bitacorasPort, 27

- compara, 9
- compara::bitacoras, 26
 - comparabitacoras, 26
- compara::bitacorasPort, 27
 - bitacorasPort, 27
 - getComparabitacoras, 27
- comparabitacoras
 - compara::bitacoras, 26
 - comparaup::comparabitacoraup, 29
- comparabitacoraupPort
 - comparaup::comparabitacoraupPort, 30
- comparaup, 10
- comparaup::comparabitacoraup, 29
 - comparabitacoras, 29
 - comparaup::comparabitacoraupPort, 30
 - comparabitacoraupPort, 30
 - getComparabitacoras, 30
- crea, 11
- crea::recibe, 40
 - creararchivo, 40
- crea::recibePort, 41
 - getCreararchivo, 41
 - recibePort, 41
- creararchivo
 - crea::recibe, 40

- descarga
 - obtienearchivo::archivo, 19

- espejoPort
 - regresa::espejoPort, 33
- existePort
 - verifica::existePort, 35

- fechaedicion
 - archivodate::fecha, 37
- fechaPort
 - archivodate::fechaPort, 38

- getArchivokb
 - valor::tamanoPort, 44
- getComparabitacoras
 - compara::bitacorasPort, 27
 - comparaup::comparabitacoraupPort, 30
- getCreararchivo
 - crea::recibePort, 41
- getDescarga
 - obtienearchivo::archivoPort, 21
- getFechaedicion
 - archivodate::fechaPort, 38

- getIgual
 - regresa::espejoPort, 33
- getLista
 - reemplazo::archivosreemplazoPort, 24
 - visualiza::todosarchivosPort, 47
- getValida
 - acceso::userPort, 50
- getVerificaarchivo
 - verifica::existePort, 35
- igual
 - regresa::espejo, 32
- lista
 - reemplazo::archivosreemplazo, 23
 - visualiza::todosarchivos, 46
- obtienearchivo, 12
- obtienearchivo::archivo, 19
 - descarga, 19
- obtienearchivo::archivoPort, 21
 - archivoPort, 21
 - getDescarga, 21
- recibePort
 - crea::recibePort, 41
- reemplazo, 13
- reemplazo::archivosreemplazo, 23
 - lista, 23
- reemplazo::archivosreemplazoPort, 24
 - archivosreemplazoPort, 24
 - getLista, 24
- regresa, 14
- regresa::espejo, 32
 - igual, 32
- regresa::espejoPort, 33
 - espejoPort, 33
 - getIgual, 33
- src/java/acceso/user.java, 53
- src/java/acceso/userPort.java, 55
- src/java/archivodate/fecha.java, 58
- src/java/archivodate/fechaPort.java, 61
- src/java/compara/bitacoras.java, 64
- src/java/compara/bitacorasPort.java, 67
- src/java/comparaup/comparabitacoraup.java, 70
- src/java/comparaup/comparabitacoraupPort.java, 73
- src/java/crea/recibe.java, 76
- src/java/crea/recibePort.java, 79
- src/java/obtienearchivo/archivo.java, 82
- src/java/obtienearchivo/archivoPort.java, 85
- src/java/reemplazo/archivosreemplazo.java, 88
- src/java/reemplazo/archivosreemplazoPort.java, 91
- src/java/regresa/espejo.java, 94
- src/java/regresa/espejoPort.java, 96
- src/java/valor/tamano.java, 99
- src/java/valor/tamanoPort.java, 102
- src/java/verifica/existe.java, 105
- src/java/verifica/existePort.java, 108
- src/java/visualiza/todosarchivos.java, 111
- src/java/visualiza/todosarchivosPort.java, 114
- tamanoPort
 - valor::tamanoPort, 44
- todosarchivosPort
 - visualiza::todosarchivosPort, 47
- userPort
 - acceso::userPort, 50
- valida
 - acceso::user, 49
- valor, 15
- valor::tamano, 43
 - archivokb, 43
- valor::tamanoPort, 44
 - getArchivokb, 44
 - tamanoPort, 44
- verifica, 16
- verifica::existe, 34
 - verificaarchivo, 34
- verifica::existePort, 35
 - existePort, 35
 - getVerificaarchivo, 35
- verificaarchivo

- verifica::existe, 34
- visualiza, 17
- visualiza::todosarchivos, 46
 - lista, 46
- visualiza::todosarchivosPort, 47
 - getLista, 47
 - todosarchivosPort, 47