Memoria de la aplicación distribuida

Sistemas y Servicios Distribuidos

Juan Luis Navarro Rey

Resumen

Esta es la memoria de la aplicación distribuida mandada en prácticas de la asignatura Sistemas y Servicios Distribuidos.
Esta memoria ha sido realizada durante el curso académico 2014/2015.

Contenidos

1 Introducción

Esta aplicación consiste en sincronizar dos directorios. Un directorio en un cliente y otro en un servidor. Si el fichero que se encuentra en el cliente no existe en el directorio del servidor, entonces el cliente subirá el fichero al servidor. Si el fichero que se encuentra en el servidor no existe en el cliente, entonces el cliente se descargará del servidor dicho fichero.

Para implementar la aplicación se va a usar java rmi. La aplicación constará de tres partes:

- Servidor
- Proxy
- Cliente

1.1 Java RMI

RMI (Java Remote Method Invocation) es un mecanismo ofrecido por Java para invocar un método de manera remota. Forma parte del entorno estándar de ejecución de Java y proporciona un mecanismo simple para la comunicación de servidores en aplicaciones distribuidas basadas exclusivamente en Java. Si se requiere comunicación entre otras tecnologías debe utilizarse CORBA o SOAP en lugar de RMI.
RMI se caracteriza por la facilidad de su uso en la programación por estar específicamente diseñado para Java; proporciona paso de objetos por referencia (no permitido por SOAP), recolección de basura distribuida (Garbage Collector distribuido) y paso de tipos arbitrarios (funcionalidad no provista por CORBA).

1.2 Servidor

Se ha diseñado el servidor como una aplicación de consola ya que no tiene sentido crear una interfaz gráfica ya que el cliente hará una petición al servidor mediante el proxy al servidor. El servidor ejecutará la petición y devolverá al cliente el resultado mediante el proxy.

1.3 Proxy

El proxy es una interfaz en la cual tiene definidas las cabeceras de las funciones del servidor. El proxy es el encargado de hacer posible la comunicación entre el cliente y el servidor dando transparencia al cliente.

1.4 Cliente

Se ha diseñado el cliente con una interfaz gráfica para simplificar el manejo de la aplicación distribuida al cliente. Cuando el cliente desea realizar una petición al servidor, éste invoca al proxy y el proxy realiza la petición al servidor. Cuando el servidor acaba, el resultado le llega al cliente mediante el proxy.

1.5 Configuración

Para configurar correctamente RMI es necesario cumplir los siguientes pasos:

1. Crear el fichero server.policy
2. Es necesario pasar parámetros a la máquina virtual java.

1.5.1 Configuración del fichero server.policy

Este fichero contiene la configuración de permisos de acceso. Para dar correctamente hay que crear el fichero server.policy con el siguiente código.

```
grant{
  permission java.security.AllPermission;
};
```

1.5.2 Configuración de la máquina virtual

Una vez creado el fichero server.policy hay que pasar los siguientes parámetros a la máquina virtual.

1. -Djava.rmi.server.codebase=file://<ruta>
2. -Djava.security.policy=file://<ruta>server.policy

En el primer parámetro se indica la ruta de los ficheros binarios de la aplicación. En el segundo parámetro se indica la ruta del fichero server.policy

2 Servidor

2.1 Introducción

Se ha diseñado el servidor como una aplicación de consola ya que no tiene sentido crear una interfaz gráfica ya que el cliente hará una petición al servidor mediante el proxy al servidor. El servidor ejecutará la petición y devolverá al cliente el resultado mediante el proxy.

2.2 Funcionalidad del servidor

El servidor tiene las siguientes funcionalidades: descargar fichero, dar la hora que tiene, lista los ficheros de la carpeta remota, cambiar el directorio remoto, subir un fichero y por último devolver la última modificación de un fichero.

2.2.1 Descargar fichero

Descarga un fichero sin tener en cuenta la fecha de la última modificación. Si se quiere tener en cuenta la fecha de la última modificación, hay que comprobar previamente que el fichero del cliente es más viejo que el del servidor.
Este método descarga un fichero mediante un buffer de bytes el cual se envía al cliente. Se realiza de esta manera para optimizar el uso de la aplicación. Los parámetros que acepta el método son los siguientes:

- name: Nombre del fichero que se quiere descargar.
- index: Se empieza a llenar el buffer, el cual se va a enviar al cliente, en el byte número index del texto que contiene el fichero que se quiere descargar.

Lo primero que hace el método es comprobar que el fichero que se quiere descargar existe en la carpeta del servidor. Si no existe se manda un mensaje de error al cliente.
A continuación se calcula el número de la iteración, es decir, cuantos buffers (partes) se le han mandado ya al cliente. Se utiliza el número de iteración para saber si en la iteración actual se va a rellenar el buffer. Si es así crea el buffer con la longitud habitual. De no rellenarse el buffer, se calcula el número de bytes que se van a mandar al cliente y se crea el buffer con dicha longitud.
Una vez rellenado el buffer se calcula si en la iteracción actual se completa el fichero. Si es así, al final del buffer se manda un carácter de escape seguido de la cadena 'FIN'. De no completarse el fichero en la iteración actual, al final del buffer se manda un carácter de escape seguido de la cadena 'continue' para que el cliente sepa que tiene que mandar una nueva petición al servidor.

2.2.2 Devolver la hora

El servidor accede al reloj del sistema operativo, captura la hora y la traduce en milisegundos. A continuación, se la envía al cliente si no se produce ningún error. En el caso de que se produzca un error en alguna operación le envía el servidor al cliente un mensaje de error.

2.2.3 Listar ficheros

Este método se encarga de averiguar el nombre de todos los ficheros que se encuentran en la carpeta remota.

2.2.4 Modificar la carpeta remota del servidor

Para implementar esta función se han generado dos métodos en el servidor. El primero, no tiene parámetros y lista los posibles directorios remotos del servidor y se los envía al cliente. El cliente elige el directorio remoto del servidor y llama al segundo método que coge como parámetro la ruta del directorio. Este segundo método, coge la ruta pasada como parámetro y le asocia la carpeta remota.

2.2.5 Subir fichero

Sube un fichero sin tener en cuenta la fecha de la última modificación. Si se quiere tener en cuenta la fecha de la última modificación, hay que comprobar previamente que el fichero del cliente es más nuevo que el del servidor. Este método sube un fichero mediante un buffer de bytes el cual se envía al servidor. Se realiza de esta manera para optimizar el uso de la aplicación. Los parámetros que acepta el método son los siguientes:

- name: Nombre del fichero que se quiere subir.
- buffer: Parte del fichero que se va a subir al servidor.
- actualizar: Vale true si se va a actualizar el fichero y por tanto hay que borrar el fichero existente.

Lo primero que hace el método es comprobar que el fichero que se quiere subir existe. Si no existiese, el servidor lo crea.
Lo siguiente es comprobar si la parte enviada es la primera parte o no. Si es la primera parte, entonces el parámetro actualizar vale true. Si no es la primera parte, entonces se lee el contenido del fichero y se guarda en una variable.
Después, se añade a la variable auxiliar el buffer de contenido enviado por el cliente.
A continuación, se comprueba que el fichero que se quiere escribir tiene permisos de escritura. Si no los tiene, se manda un mensaje de error al cliente. Si se puede escribir, entonces se escribe en el fichero.

2.2.6 Última modificación

Averigua la última modificación de un fichero. Este método acepta el siguiente parámetro:

- nombre: Nombre del fichero que se quiere saber la última modificación.

Lo primero que hace es comprobar si el fichero existe. Si el fichero no existe, se manda un mensaje de error al cliente. Si existe el fichero, se calculará la última modificación y se enviará al cliente.

3 Proxy

3.1 Introducción

El proxy es una interfaz en la cual tiene definidas las cabeceras de las funciones del servidor. El proxy es el encargado de hacer posible la comunicación entre el cliente y el servidor dando transparencia al cliente.

3.2 Funcionalidad

El proxy contiene las siguientes funciones (métodos):

Función	Descripción
descargarFichero(java.lang.String name, int index)	Sube un fichero a la carpeta remota.
getHoraServidor()	El servidor calcula la hora que tiene y la devuelve en milisegundos
listaFicherosCarpetaRemota()	Crea una lista con el nombre de todos los ficheros que existen el la carpeta remota del servidor
modificaCarpetaServidor()	Modifica la carpeta remota del servidor
modificaCarpetaServidor(java.lang.String carpetaServidor)	Modifica la carpeta remota del servidor
subirFichero(java.lang.String name, byte[] contenido,boolean actualizar)	Sube un fichero a la carpeta remota
ultimaModificacion(java.lang.String nombre)	Averigua la ultima modificacion de un fichero

Tabla 3.1: *Funcionalidad del proxy*

Para ver la documentación con más detalle, ver la sección A.2.

4 Cliente

4.1 Introducción

Se ha diseñado el cliente con una interfaz gráfica para simplificar el manejo de la aplicación distribuida al cliente. Cuando el cliente desea realizar una petición al servidor, éste invoca al proxy y el proxy realiza la petición al servidor. Cuando el servidor acaba, el resultado le llega al cliente mediante el proxy.

4.2 Ejecución de la aplicación

Una vez que se ejecuta la aplicación, lo primero que pide es la IP del servidor tal y como se ve en la figura 4.1 Una vez introducida la IP se ejecuta la ventana principal de la aplicación.

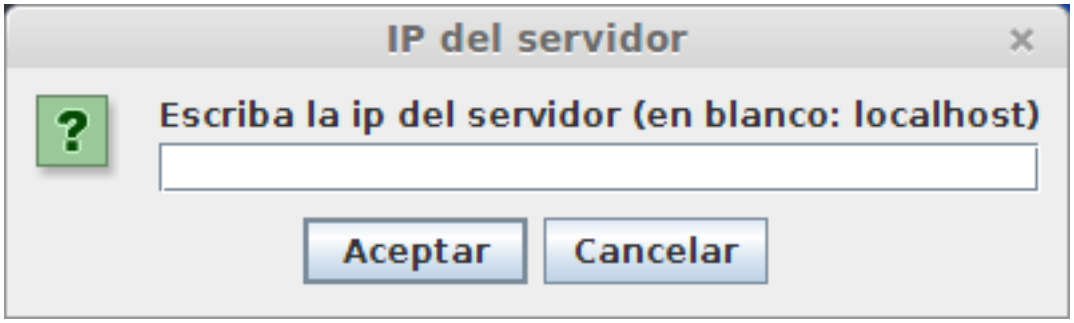

Figura 4.1: *Elegir IP*

4.2.1 Ventana principal

La ventana principal (ver figura 4.2) consta de una barra de menú y de cuatro botones:

- Subir fichero: sube un fichero al servidor.
- Descargar fichero: descarga un fichero al cliente.
- Sincronizar reloj: Sincroniza el reloj del cliente con el del servidor.
- Actualizar directorios: Actualiza los directorios (remoto y local).

El contenido de la barra de menú es el siguiente:

- Archivo
  - Ajuestes
    - Cambiar carpeta local: cambia la carpeta local

Figura 4.2: *Ventana principal*

  - ∗ Cambiar carpeta remota: cambia la carpeta remota (ver figura 4.3)
  - ∗ Cambiar Tmin: cambia el tiempo mínimo de transmisión de un paquete del cliente al servidor
- – Salir: cierra la aplicación

- About
  - – Carpeta local: muestra la ruta de la carpeta local.
  - – Tmin: muestra el valor del tiempo mínimo de transmisión de un paquete del cliente al servidor
  - – Información: muestra la información acerca de la aplicación

Figura 4.3: *Elegir directorio remoto*

4.2.2 Elegir directorio remoto

Como se puede observar en la figura 4.3, la ventana consta de dos partes.
La primera parte contiene un desplegable con todas las posibles carpetas (existentes) del servidor que pueden ser la nueva carpeta remota.
Si por algún motivo se quisiese elegir otra carpeta en la segunda parte se ha creado un campo de texto para introducir la ruta.
Cuando se le da al botón de aceptar, por defecto escoge la ruta de la primera parte. Si se quiere poner una ruta manualmente es necesario activar la casilla que hay justo encima para que la ruta que se pase al servidor sea la ruta manual.

A

Documentación de la aplicación distribuida

A.1 Javadoc

Javadoc es una utilidad de Oracle para la generación de documentación de APIs en formato HTML a partir de código fuente Java. Javadoc es el estándar de la industria para documentar clases de Java. La mayoría de los IDEs los generan automáticamente.

A continuación se explican algunas de las palabras reservadas:

Tag	Descripción
@author	Nombre del desarrollador
@deprecated	Indica que el método o clase es antigua y que no se recomienda su uso porque posiblemente desaparecerá en versiones posteriores
@param	Definición de un parámetro de un método, es requerido para todos los parámetros del método
@return	Informa de lo que devuelve el método, no se puede usar en constructores o métodos "void"
@see	Asocia con otro método o clase
@throws	Excepción lanzada por el método
@version	Versión del método o clase

Tabla A.1: *Uso de los tags*

A.2 Documentación

En las siguientes páginas se puede ver la documentación de la aplicación.

Class Cliente

java.lang.Object
    Cliente

```
public class Cliente
extends java.lang.Object
```

Clase principal del cliente.

Version:

1.0

Author:

Juan Luis Navarro Rey.

Constructor Summary

Constructors

Constructor and Description
`Cliente()`

Method Summary

Methods

Modifier and Type	Method and Description
static void	`main(java.lang.String[] args)`

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Detail

Cliente

```
public Cliente()
```

Method Detail

main

```
public static void main(java.lang.String[] args)
                 throws java.rmi.RemoteException,
                        java.rmi.NotBoundException,
                        java.net.MalformedURLException
```

Throws:

`java.rmi.RemoteException`

`java.rmi.NotBoundException`

```
java.net.MalformedURLException
```

Package Class Tree Deprecated Index Help

Prev Class Next Class Frames No Frames All Classes

Summary: Nested | Field | Constr | Method Detail: Field | Constr | Method

Class GuiCliente

java.lang.Object
  java.awt.Component
    java.awt.Container
      java.awt.Window
        java.awt.Frame
          javax.swing.JFrame
            GuiCliente

All Implemented Interfaces:

java.awt.image.ImageObserver, java.awt.MenuContainer, java.io.Serializable, javax.accessibility.Accessible, javax.swing.RootPaneContainer, javax.swing.WindowConstants

---

```
public class GuiCliente
extends javax.swing.JFrame
```

Interfaz grafica que interactua con el usuario

Version:

1.0

Author:

Juan Luis Navarro Rey

See Also:

Serialized Form

Nested Class Summary

Nested classes/interfaces inherited from class javax.swing.JFrame

`javax.swing.JFrame.AccessibleJFrame`

Nested classes/interfaces inherited from class java.awt.Frame

`java.awt.Frame.AccessibleAWTFrame`

Nested classes/interfaces inherited from class java.awt.Window

`java.awt.Window.AccessibleAWTWindow, java.awt.Window.Type`

Nested classes/interfaces inherited from class java.awt.Container

`java.awt.Container.AccessibleAWTContainer`

Nested classes/interfaces inherited from class java.awt.Component

`java.awt.Component.AccessibleAWTComponent, java.awt.Component.BaselineResizeBehavior, java.awt.Component.BltBufferStrategy, java.awt.Component.FlipBufferStrategy`

Field Summary

Fields inherited from class javax.swing.JFrame

`accessibleContext, EXIT_ON_CLOSE, rootPane, rootPaneCheckingEnabled`

Fields inherited from class java.awt.Frame

CROSSHAIR_CURSOR, DEFAULT_CURSOR, E_RESIZE_CURSOR, HAND_CURSOR, ICONIFIED, MAXIMIZED_BOTH, MAXIMIZED_HORIZ, MAXIMIZED_VERT, MOVE_CURSOR, N_RESIZE_CURSOR, NE_RESIZE_CURSOR, NORMAL, NW_RESIZE_CURSOR, S_RESIZE_CURSOR, SE_RESIZE_CURSOR, SW_RESIZE_CURSOR, TEXT_CURSOR, W_RESIZE_CURSOR, WAIT_CURSOR

Fields inherited from class java.awt.Component

BOTTOM_ALIGNMENT, CENTER_ALIGNMENT, LEFT_ALIGNMENT, RIGHT_ALIGNMENT, TOP_ALIGNMENT

Fields inherited from interface javax.swing.WindowConstants

DISPOSE_ON_CLOSE, DO_NOTHING_ON_CLOSE, HIDE_ON_CLOSE

Fields inherited from interface java.awt.image.ImageObserver

ABORT, ALLBITS, ERROR, FRAMEBITS, HEIGHT, PROPERTIES, SOMEBITS, WIDTH

Constructor Summary

Constructors

Constructor and Description
GuiCliente(Proxy proxy) Constructor de la clase.

Method Summary

Methods inherited from class javax.swing.JFrame

addImpl, createRootPane, frameInit, getAccessibleContext, getContentPane, getDefaultCloseOperation, getGlassPane, getGraphics, getJMenuBar, getLayeredPane, getRootPane, getTransferHandler, isDefaultLookAndFeelDecorated, isRootPaneCheckingEnabled, paramString, processWindowEvent, remove, repaint, setContentPane, setDefaultCloseOperation, setDefaultLookAndFeelDecorated, setGlassPane, setIconImage, setJMenuBar, setLayeredPane, setLayout, setRootPane, setRootPaneCheckingEnabled, setTransferHandler, update

Methods inherited from class java.awt.Frame

addNotify, getCursorType, getExtendedState, getFrames, getIconImage, getMaximizedBounds, getMenuBar, getState, getTitle, isResizable, isUndecorated, remove, removeNotify, setBackground, setCursor, setExtendedState, setMaximizedBounds, setMenuBar, setOpacity, setResizable, setShape, setState, setTitle, setUndecorated

Methods inherited from class java.awt.Window

addPropertyChangeListener, addPropertyChangeListener, addWindowFocusListener, addWindowListener, addWindowStateListener, applyResourceBundle, applyResourceBundle, createBufferStrategy, createBufferStrategy, dispose, getBackground, getBufferStrategy, getFocusableWindowState, getFocusCycleRootAncestor, getFocusOwner, getFocusTraversalKeys, getIconImages, getInputContext, getListeners, getLocale, getModalExclusionType, getMostRecentFocusOwner, getOpacity, getOwnedWindows, getOwner, getOwnerlessWindows, getShape, getToolkit, getType, getWarningString, getWindowFocusListeners, getWindowListeners, getWindows, getWindowStateListeners, hide, isActive, isAlwaysOnTop, isAlwaysOnTopSupported, isAutoRequestFocus, isFocusableWindow, isFocusCycleRoot, isFocused, isLocationByPlatform, isOpaque, isShowing, isValidateRoot, pack, paint, postEvent, processEvent, processWindowFocusEvent, processWindowStateEvent, removeWindowFocusListener, removeWindowListener, removeWindowStateListener, reshape, setAlwaysOnTop, setAutoRequestFocus, setBounds, setBounds, setCursor, setFocusableWindowState, setFocusCycleRoot, setIconImages, setLocation, setLocation, setLocationByPlatform, setLocationRelativeTo, setMinimumSize, setModalExclusionType, setSize, setSize, setType, setVisible, show, toBack, toFront

Methods inherited from class java.awt.Container

add, add, add, add, add, addContainerListener, applyComponentOrientation, areFocusTraversalKeysSet, countComponents, deliverEvent, doLayout, findComponentAt, findComponentAt, getAlignmentX, getAlignmentY, getComponent, getComponentAt, getComponentAt, getComponentCount, getComponents, getComponentZOrder, getContainerListeners, getFocusTraversalPolicy, getInsets, getLayout, getMaximumSize, getMinimumSize, getMousePosition, getPreferredSize, insets, invalidate, isAncestorOf, isFocusCycleRoot, isFocusTraversalPolicyProvider, isFocusTraversalPolicySet, layout, list, list, locate, minimumSize, paintComponents, preferredSize, print, printComponents,

processContainerEvent, remove, removeAll, removeContainerListener, setComponentZOrder, setFocusTraversalKeys, setFocusTraversalPolicy, setFocusTraversalPolicyProvider, setFont, transferFocusDownCycle, validate, validateTree

Methods inherited from class java.awt.Component

action, add, addComponentListener, addFocusListener, addHierarchyBoundsListener, addHierarchyListener, addInputMethodListener, addKeyListener, addMouseListener, addMouseMotionListener, addMouseWheelListener, bounds, checkImage, checkImage, coalesceEvents, contains, contains, createImage, createImage, createVolatileImage, createVolatileImage, disable, disableEvents, dispatchEvent, enable, enable, enableEvents, enableInputMethods, firePropertyChange, firePropertyChange, firePropertyChange, firePropertyChange, firePropertyChange, firePropertyChange, firePropertyChange, firePropertyChange, firePropertyChange, getBaseline, getBaselineResizeBehavior, getBounds, getBounds, getColorModel, getComponentListeners, getComponentOrientation, getCursor, getDropTarget, getFocusListeners, getFocusTraversalKeysEnabled, getFont, getFontMetrics, getForeground, getGraphicsConfiguration, getHeight, getHierarchyBoundsListeners, getHierarchyListeners, getIgnoreRepaint, getInputMethodListeners, getInputMethodRequests, getKeyListeners, getLocation, getLocation, getLocationOnScreen, getMouseListeners, getMouseMotionListeners, getMousePosition, getMouseWheelListeners, getName, getParent, getPeer, getPropertyChangeListeners, getPropertyChangeListeners, getSize, getSize, getTreeLock, getWidth, getX, getY, gotFocus, handleEvent, hasFocus, imageUpdate, inside, isBackgroundSet, isCursorSet, isDisplayable, isDoubleBuffered, isEnabled, isFocusable, isFocusOwner, isFocusTraversable, isFontSet, isForegroundSet, isLightweight, isMaximumSizeSet, isMinimumSizeSet, isPreferredSizeSet, isValid, isVisible, keyDown, keyUp, list, list, list, location, lostFocus, mouseDown, mouseDrag, mouseEnter, mouseExit, mouseMove, mouseUp, move, nextFocus, paintAll, prepareImage, prepareImage, printAll, processComponentEvent, processFocusEvent, processHierarchyBoundsEvent, processHierarchyEvent, processInputMethodEvent, processKeyEvent, processMouseEvent, processMouseMotionEvent, processMouseWheelEvent, removeComponentListener, removeFocusListener, removeHierarchyBoundsListener, removeHierarchyListener, removeInputMethodListener, removeKeyListener, removeMouseListener, removeMouseMotionListener, removeMouseWheelListener, removePropertyChangeListener, removePropertyChangeListener, repaint, repaint, repaint, requestFocus, requestFocus, requestFocusInWindow, requestFocusInWindow, resize, resize, revalidate, setComponentOrientation, setDropTarget, setEnabled, setFocusable, setFocusTraversalKeysEnabled, setForeground, setIgnoreRepaint, setLocale, setMaximumSize, setName, setPreferredSize, show, size, toString, transferFocus, transferFocusBackward, transferFocusUpCycle

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait

Methods inherited from interface java.awt.MenuContainer

getFont, postEvent

Constructor Detail

GuiCliente

```
public GuiCliente(Proxy proxy)
           throws java.rmi.RemoteException,
                  java.rmi.NotBoundException,
                  java.net.MalformedURLException
```

Constructor de la clase.

Parameters:

`proxy` - Interfaz que conectara el cliente con el servidor.

Throws:

`java.rmi.RemoteException`

`java.rmi.NotBoundException`

`java.net.MalformedURLException`

Class ElegirCarpetaRemota

java.lang.Object
  java.awt.Component
    java.awt.Container
      java.awt.Window
        java.awt.Frame
          javax.swing.JFrame
            ElegirCarpetaRemota

All Implemented Interfaces:

java.awt.image.ImageObserver, java.awt.MenuContainer, java.io.Serializable, javax.accessibility.Accessible, javax.swing.RootPaneContainer, javax.swing.WindowConstants

---

```
public class ElegirCarpetaRemota
extends javax.swing.JFrame
```

Clase auxiliar para elegir carpeta remota.

Version:

1.0

Author:

Juan Luis Navarro Rey.

See Also:

Serialized Form

Nested Class Summary

Nested classes/interfaces inherited from class javax.swing.JFrame

`javax.swing.JFrame.AccessibleJFrame`

Nested classes/interfaces inherited from class java.awt.Frame

`java.awt.Frame.AccessibleAWTFrame`

Nested classes/interfaces inherited from class java.awt.Window

`java.awt.Window.AccessibleAWTWindow, java.awt.Window.Type`

Nested classes/interfaces inherited from class java.awt.Container

`java.awt.Container.AccessibleAWTContainer`

Nested classes/interfaces inherited from class java.awt.Component

`java.awt.Component.AccessibleAWTComponent, java.awt.Component.BaselineResizeBehavior, java.awt.Component.BltBufferStrategy, java.awt.Component.FlipBufferStrategy`

Field Summary

Fields inherited from class javax.swing.JFrame

`accessibleContext, EXIT_ON_CLOSE, rootPane, rootPaneCheckingEnabled`

Fields inherited from class java.awt.Frame

CROSSHAIR_CURSOR, DEFAULT_CURSOR, E_RESIZE_CURSOR, HAND_CURSOR, ICONIFIED, MAXIMIZED_BOTH, MAXIMIZED_HORIZ, MAXIMIZED_VERT, MOVE_CURSOR, N_RESIZE_CURSOR, NE_RESIZE_CURSOR, NORMAL, NW_RESIZE_CURSOR, S_RESIZE_CURSOR, SE_RESIZE_CURSOR, SW_RESIZE_CURSOR, TEXT_CURSOR, W_RESIZE_CURSOR, WAIT_CURSOR

Fields inherited from class java.awt.Component

BOTTOM_ALIGNMENT, CENTER_ALIGNMENT, LEFT_ALIGNMENT, RIGHT_ALIGNMENT, TOP_ALIGNMENT

Fields inherited from interface javax.swing.WindowConstants

DISPOSE_ON_CLOSE, DO_NOTHING_ON_CLOSE, HIDE_ON_CLOSE

Fields inherited from interface java.awt.image.ImageObserver

ABORT, ALLBITS, ERROR, FRAMEBITS, HEIGHT, PROPERTIES, SOMEBITS, WIDTH

Constructor Summary

Constructors

Constructor and Description
ElegirCarpetaRemota(javax.swing.ComboBoxModel<java.lang.String> model, Proxy proxy) Constructor de la clase.

Method Summary

Methods inherited from class javax.swing.JFrame

addImpl, createRootPane, frameInit, getAccessibleContext, getContentPane, getDefaultCloseOperation, getGlassPane, getGraphics, getJMenuBar, getLayeredPane, getRootPane, getTransferHandler, isDefaultLookAndFeelDecorated, isRootPaneCheckingEnabled, paramString, processWindowEvent, remove, repaint, setContentPane, setDefaultCloseOperation, setDefaultLookAndFeelDecorated, setGlassPane, setIconImage, setJMenuBar, setLayeredPane, setLayout, setRootPane, setRootPaneCheckingEnabled, setTransferHandler, update

Methods inherited from class java.awt.Frame

addNotify, getCursorType, getExtendedState, getFrames, getIconImage, getMaximizedBounds, getMenuBar, getState, getTitle, isResizable, isUndecorated, remove, removeNotify, setBackground, setCursor, setExtendedState, setMaximizedBounds, setMenuBar, setOpacity, setResizable, setShape, setState, setTitle, setUndecorated

Methods inherited from class java.awt.Window

addPropertyChangeListener, addPropertyChangeListener, addWindowFocusListener, addWindowListener, addWindowStateListener, applyResourceBundle, applyResourceBundle, createBufferStrategy, createBufferStrategy, dispose, getBackground, getBufferStrategy, getFocusableWindowState, getFocusCycleRootAncestor, getFocusOwner, getFocusTraversalKeys, getIconImages, getInputContext, getListeners, getLocale, getModalExclusionType, getMostRecentFocusOwner, getOpacity, getOwnedWindows, getOwner, getOwnerlessWindows, getShape, getToolkit, getType, getWarningString, getWindowFocusListeners, getWindowListeners, getWindows, getWindowStateListeners, hide, isActive, isAlwaysOnTop, isAlwaysOnTopSupported, isAutoRequestFocus, isFocusableWindow, isFocusCycleRoot, isFocused, isLocationByPlatform, isOpaque, isShowing, isValidateRoot, pack, paint, postEvent, processEvent, processWindowFocusEvent, processWindowStateEvent, removeWindowFocusListener, removeWindowListener, removeWindowStateListener, reshape, setAlwaysOnTop, setAutoRequestFocus, setBounds, setBounds, setCursor, setFocusableWindowState, setFocusCycleRoot, setIconImages, setLocation, setLocation, setLocationByPlatform, setLocationRelativeTo, setMinimumSize, setModalExclusionType, setSize, setSize, setType, setVisible, show, toBack, toFront

Methods inherited from class java.awt.Container

add, add, add, add, add, addContainerListener, applyComponentOrientation, areFocusTraversalKeysSet, countComponents, deliverEvent, doLayout, findComponentAt, findComponentAt, getAlignmentX, getAlignmentY, getComponent, getComponentAt, getComponentAt, getComponentCount, getComponents, getComponentZOrder, getContainerListeners, getFocusTraversalPolicy, getInsets, getLayout, getMaximumSize, getMinimumSize, getMousePosition, getPreferredSize, insets, invalidate, isAncestorOf, isFocusCycleRoot, isFocusTraversalPolicyProvider, isFocusTraversalPolicySet, layout, list, list, locate, minimumSize, paintComponents, preferredSize, print, printComponents,

processContainerEvent, remove, removeAll, removeContainerListener, setComponentZOrder, setFocusTraversalKeys, setFocusTraversalPolicy, setFocusTraversalPolicyProvider, setFont, transferFocusDownCycle, validate, validateTree

Methods inherited from class java.awt.Component

action, add, addComponentListener, addFocusListener, addHierarchyBoundsListener, addHierarchyListener, addInputMethodListener, addKeyListener, addMouseListener, addMouseMotionListener, addMouseWheelListener, bounds, checkImage, checkImage, coalesceEvents, contains, contains, createImage, createImage, createVolatileImage, createVolatileImage, disable, disableEvents, dispatchEvent, enable, enable, enableEvents, enableInputMethods, firePropertyChange, firePropertyChange, firePropertyChange, firePropertyChange, firePropertyChange, firePropertyChange, firePropertyChange, firePropertyChange, firePropertyChange, getBaseline, getBaselineResizeBehavior, getBounds, getBounds, getColorModel, getComponentListeners, getComponentOrientation, getCursor, getDropTarget, getFocusListeners, getFocusTraversalKeysEnabled, getFont, getFontMetrics, getForeground, getGraphicsConfiguration, getHeight, getHierarchyBoundsListeners, getHierarchyListeners, getIgnoreRepaint, getInputMethodListeners, getInputMethodRequests, getKeyListeners, getLocation, getLocation, getLocationOnScreen, getMouseListeners, getMouseMotionListeners, getMousePosition, getMouseWheelListeners, getName, getParent, getPeer, getPropertyChangeListeners, getPropertyChangeListeners, getSize, getSize, getTreeLock, getWidth, getX, getY, gotFocus, handleEvent, hasFocus, imageUpdate, inside, isBackgroundSet, isCursorSet, isDisplayable, isDoubleBuffered, isEnabled, isFocusable, isFocusOwner, isFocusTraversable, isFontSet, isForegroundSet, isLightweight, isMaximumSizeSet, isMinimumSizeSet, isPreferredSizeSet, isValid, isVisible, keyDown, keyUp, list, list, list, location, lostFocus, mouseDown, mouseDrag, mouseEnter, mouseExit, mouseMove, mouseUp, move, nextFocus, paintAll, prepareImage, prepareImage, printAll, processComponentEvent, processFocusEvent, processHierarchyBoundsEvent, processHierarchyEvent, processInputMethodEvent, processKeyEvent, processMouseEvent, processMouseMotionEvent, processMouseWheelEvent, removeComponentListener, removeFocusListener, removeHierarchyBoundsListener, removeHierarchyListener, removeInputMethodListener, removeKeyListener, removeMouseListener, removeMouseMotionListener, removeMouseWheelListener, removePropertyChangeListener, removePropertyChangeListener, repaint, repaint, repaint, requestFocus, requestFocus, requestFocusInWindow, requestFocusInWindow, resize, resize, revalidate, setComponentOrientation, setDropTarget, setEnabled, setFocusable, setFocusTraversalKeysEnabled, setForeground, setIgnoreRepaint, setLocale, setMaximumSize, setName, setPreferredSize, show, size, toString, transferFocus, transferFocusBackward, transferFocusUpCycle

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait

Methods inherited from interface java.awt.MenuContainer

getFont, postEvent

Constructor Detail

ElegirCarpetaRemota

```
public ElegirCarpetaRemota(javax.swing.ComboBoxModel<java.lang.String> model,
                           Proxy proxy)
```

Constructor de la clase.

Parameters:

`model` - Lista con el nombre de las posibles carpetas remotas.

`proxy` - Interfaz que conecta el cliente con el servidor.

Package Class Tree Deprecated Index Help

Prev Class Next Class Frames No Frames All Classes

Summary: Nested | Field | Constr | Method Detail: Field | Constr | Method

Package Class Use Tree Deprecated Index Help

Prev Class Next Class Frames No Frames All Classes

Summary: Nested | Field | Constr | Method Detail: Field | Constr | Method

Interface Proxy

All Superinterfaces:

java.rmi.Remote

All Known Implementing Classes:

Servidor

```
public interface Proxy
extends java.rmi.Remote
```

Interfaz intermediaria entre el cliente y el servidor.

Version:

2.0

Author:

Juan Luis Navarro Rey.

Method Summary

Methods

Modifier and Type	Method and Description
RespuestaServidor<byte[]>	descargarFichero(java.lang.String name, int index) Descarga un fichero a la carpeta local.
RespuestaServidor<java.lang.Long>	getHoraServidor() El servidor calcula la hora que tiene y la devuelve en milisegundos.
RespuestaServidor<java.util.List<java.lang.String>>	listaFicherosCarpetaRemota() Crea una lista con el nombre de todos los ficheros que existen el la carpeta remota del servidor.
RespuestaServidor<javax.swing.DefaultComboBoxModel<java.lang.String>>	modificaCarpetaServidor() Modifica la carpeta remota del servidor.
RespuestaServidor<java.lang.Boolean>	modificaCarpetaServidor(java.lang.String carpetaServidor) Modifica la carpeta remota del servidor.
RespuestaServidor<java.lang.Boolean>	subirFichero(java.lang.String name, byte[] contenido, boolean actualizar) Sube un fichero a la carpeta remota.
RespuestaServidor<java.lang.Long>	ultimaModificacion(java.lang.String nombre) Averigua la ultima modificacion de un fichero.

Method Detail

subirFichero

```
RespuestaServidor<java.lang.Boolean> subirFichero(java.lang.String name,
                                                  byte[] contenido,
                                                  boolean actualizar)
                                                    throws java.rmi.RemoteException
```

Sube un fichero a la carpeta remota.

Parameters:

`name` - Nombre del fichero que se quiere mandar.

`contenido` - Contenido o parte del contenido del fichero.

`actualizar` - Vale true si se va a actualizar el fichero y por tanto hay que borrar el fichero existente.

Returns:

Devuelve true si el envio es correcto y false en caso contrario.

Throws:

`java.rmi.RemoteException` - Lanza un error si se pierde la conexion o el fallo en el envio es grave.

descargarFichero

```
RespuestaServidor<byte[]> descargarFichero(java.lang.String name,
                                           int index)
                                             throws java.rmi.RemoteException
```

Descarga un fichero a la carpeta local.

Parameters:

`name` - Nombre del fichero que se quiere mandar.

`index` - Numero de byte en el cual el servidor va a empezar a mandar el fichero.

Returns:

Devuelve el contenido o parte del contenido del fichero.

Throws:

`java.rmi.RemoteException` - Lanza un error si se pierde la conexion o el fallo en el envio es grave.

modificaCarpetaServidor

```
RespuestaServidor<javax.swing.DefaultComboBoxModel<java.lang.String>> modificaCarpetaServidor()
                                                                                    throws java.rmi.RemoteException
```

Modifica la carpeta remota del servidor.

Returns:

Devuelve varios combobox para poder elegir la carpeta.

Throws:

`java.rmi.RemoteException` - Lanza un error si se pierde la conexion.

modificaCarpetaServidor

```
RespuestaServidor<java.lang.Boolean> modificaCarpetaServidor(java.lang.String carpetaServidor)
                                                               throws java.rmi.RemoteException
```

Modifica la carpeta remota del servidor.

Parameters:

`carpetaServidor` - Nueva carpeta remota.

Returns:

Devuelve true si se cambia correctamente la carpeta remota y false en caso contrario.

Throws:

`java.rmi.RemoteException` - Lanza un error si se pierde la conexion.

listaFicherosCarpetaRemota

```
RespuestaServidor<java.util.List<java.lang.String>> listaFicherosCarpetaRemota()
                                                                      throws java.rmi.RemoteException
```

Crea una lista con el nombre de todos los ficheros que existen el la carpeta remota del servidor.

Returns:

Devuelve una lista con el nombre de los ficheros.

Throws:

`java.rmi.RemoteException` - Lanza un error si se pierde la conexion.

ultimaModificacion

```
RespuestaServidor<java.lang.Long> ultimaModificacion(java.lang.String nombre)
                                                       throws java.rmi.RemoteException
```

Averigua la ultima modificacion de un fichero.

Parameters:

`nombre` - Nombre del fichero.

Returns:

Devuelve el tiempo de la ultima modificacion del fichero.

Throws:

`java.rmi.RemoteException` - Lanza un error si se pierde la conexion.

getHoraServidor

`RespuestaServidor<java.lang.Long> getHoraServidor()`
`throws java.rmi.RemoteException`

El servidor calcula la hora que tiene y la devuelve en milisegundos.

Returns:

Devuelve la hora en milisegundos.

Throws:

`java.rmi.RemoteException` - Lanza un error si se pierde la conexion.

Package Class Use Tree Deprecated Index Help

Prev Class Next Class Frames No Frames All Classes

Summary: Nested | Field | Constr | Method Detail: Field | Constr | Method

Class Servidor

java.lang.Object
  java.rmi.server.RemoteObject
    java.rmi.server.RemoteServer
      java.rmi.server.UnicastRemoteObject
        Servidor

All Implemented Interfaces:

java.io.Serializable, java.rmi.Remote, Proxy

```
public class Servidor
extends java.rmi.server.UnicastRemoteObject
implements Proxy
```

Implementacion del servidor.

Version:

2.0

Author:

Juan Luis Navarro Rey.

See Also:

Serialized Form

Field Summary

Fields inherited from class java.rmi.server.RemoteObject

ref

Constructor Summary

Constructors

Modifier	Constructor and Description
protected	Servidor() Inicializacion del servidor.

Method Summary

Methods

Modifier and Type	Method and Description
RespuestaServidor<byte[]>	descargarFichero(java.lang.String name, int index) Descarga un fichero a la carpeta local.
RespuestaServidor<java.lang.Long>	getHoraServidor() El servidor calcula la hora que tiene y la devuelve en milisegundos.
RespuestaServidor<java.util.List<java.lang.String>>	listaFicherosCarpetaRemota() Crea una lista con el nombre de todos los ficheros que existen el la carpeta remota del servidor.
static void	main(java.lang.String[] args)
RespuestaServidor<javax.swing.DefaultComboBoxModel<java.lang.String>>	modificaCarpetaServidor() Modifica la carpeta remota del servidor.
RespuestaServidor<java.lang.Boolean>	modificaCarpetaServidor(java.lang.String carpetaServidor) Modifica la carpeta remota del servidor.
RespuestaServidor<java.lang.Boolean>	subirFichero(java.lang.String name, byte[] contenido, boolean actualizar) Sube un fichero a la carpeta remota.
RespuestaServidor<java.lang.Long>	ultimaModificacion(java.lang.String nombre) Averigua la ultima modificacion de un fichero.

Methods inherited from class java.rmi.server.UnicastRemoteObject

clone, exportObject, exportObject, exportObject, unexportObject

Methods inherited from class java.rmi.server.RemoteServer

getClientHost, getLog, setLog

Methods inherited from class java.rmi.server.RemoteObject

equals, getRef, hashCode, toString, toStub

Methods inherited from class java.lang.Object

finalize, getClass, notify, notifyAll, wait, wait, wait

Constructor Detail

Servidor

```
protected Servidor()
            throws java.rmi.RemoteException
```

Inicializacion del servidor.

Throws:

`java.rmi.RemoteException` - Lanza un error si se pierde la conexion o el fallo en el envio es grave.

Method Detail

subirFichero

```
public RespuestaServidor<java.lang.Boolean> subirFichero(java.lang.String name,
                                                         byte[] contenido,
                                                         boolean actualizar)
                                                           throws java.rmi.RemoteException
```

Sube un fichero a la carpeta remota.

Specified by:

`subirFichero` in interface `Proxy`

Parameters:

`name` - Nombre del fichero que se quiere mandar.

`contenido` - Contenido o parte del contenido del fichero.

`actualizar` - Vale true si se va a actualizar el fichero y por tanto hay que borrar el fichero existente.

Returns:

Devuelve true si el envio es correcto y false en caso contrario.

Throws:

`java.rmi.RemoteException` - Lanza un error si se pierde la conexion o el fallo en el envio es grave.

descargarFichero

```
public RespuestaServidor<byte[]> descargarFichero(java.lang.String name,
                                                  int index)
                                                    throws java.rmi.RemoteException
```

Descarga un fichero a la carpeta local.

Specified by:

`descargarFichero` in interface `Proxy`

Parameters:

`name` - Nombre del fichero que se quiere mandar.

`index` - Numero de byte en el cual el servidor va a empezar a mandar el fichero.

Returns:

Devuelve el contenido o parte del contenido del fichero.

Throws:

java.rmi.RemoteException - Lanza un error si se pierde la conexion o el fallo en el envio es grave.

modificaCarpetaServidor

```
public RespuestaServidor<javax.swing.DefaultComboBoxModel<java.lang.String>> modificaCarpetaServidor()
                                                                                                  throws java.rmi.RemoteException
```

Modifica la carpeta remota del servidor.

Specified by:

modificaCarpetaServidor in interface Proxy

Returns:

Devuelve varios combobox para poder elegir la carpeta.

Throws:

java.rmi.RemoteException - Lanza un error si se pierde la conexion.

modificaCarpetaServidor

```
public RespuestaServidor<java.lang.Boolean> modificaCarpetaServidor(java.lang.String carpetaServidor)
                                                                 throws java.rmi.RemoteException
```

Modifica la carpeta remota del servidor.

Specified by:

modificaCarpetaServidor in interface Proxy

Parameters:

carpetaServidor - Nueva carpeta remota.

Returns:

Devuelve true si se cambia correctamente la carpeta remota y false en caso contrario.

Throws:

java.rmi.RemoteException - Lanza un error si se pierde la conexion.

listaFicherosCarpetaRemota

```
public RespuestaServidor<java.util.List<java.lang.String>> listaFicherosCarpetaRemota()
                                                                                throws java.rmi.RemoteException
```

Crea una lista con el nombre de todos los ficheros que existen el la carpeta remota del servidor.

Specified by:

listaFicherosCarpetaRemota in interface Proxy

Returns:

Devuelve una lista con el nombre de los ficheros.

Throws:

java.rmi.RemoteException - Lanza un error si se pierde la conexion.

getHoraServidor

```
public RespuestaServidor<java.lang.Long> getHoraServidor()
                                          throws java.rmi.RemoteException
```

El servidor calcula la hora que tiene y la devuelve en milisegundos.

Specified by:

getHoraServidor in interface Proxy

Returns:

Devuelve la hora en milisegundos.

Throws:

java.rmi.RemoteException - Lanza un error si se pierde la conexion.

ultimaModificacion

```
public RespuestaServidor<java.lang.Long> ultimaModificacion(java.lang.String nombre)
                                          throws java.rmi.RemoteException
```

Averigua la ultima modificacion de un fichero.

Specified by:

`ultimaModificacion` in interface `Proxy`

Parameters:

`nombre` - Nombre del fichero.

Returns:

Devuelve el tiempo de la ultima modificacion del fichero.

Throws:

`java.rmi.RemoteException` - Lanza un error si se pierde la conexion.

main

```
public static void main(java.lang.String[] args)
                throws java.rmi.RemoteException,
                       java.net.MalformedURLException
```

Throws:

`java.rmi.RemoteException`

`java.net.MalformedURLException`

Package Class Use Tree Deprecated Index Help

Prev Class Next Class Frames No Frames All Classes

Summary: Nested | Field | Constr | Method Detail: Field | Constr | Method

Class RespuestaServidor<T>

java.lang.Object
    RespuestaServidor<T>

All Implemented Interfaces:

java.io.Serializable

```
public class RespuestaServidor<T>
extends java.lang.Object
implements java.io.Serializable
```

Clase generica que contendra el valor de la respuesta del servidor y se le asocia un mensaje de error en caso de haberse producido un error.

See Also:

Serialized Form

Constructor Summary

Constructors

Constructor and Description
`RespuestaServidor(T t, java.lang.String errorMsg)` Constructor de la clase.

Method Summary

Methods

Modifier and Type	Method and Description
`java.lang.String`	`getError()` Obtiene el valor del mensaje de error.
`T`	`getValor()` Obtiene el valor del objeto genÃ©rico.

Methods inherited from class java.lang.Object

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Constructor Detail

RespuestaServidor

```
public RespuestaServidor(T t,
                         java.lang.String errorMsg)
```

Constructor de la clase.

Parameters:

`t` - Objeto genÃ©rico que se quiere inicializar.

`errorMsg` - Mensaje de error.

Method Detail

getValor

```
public T getValor()
```

Obtiene el valor del objeto genÃ©rico.

Returns:

devuelve el valor del objeto genÃ©rico.

getError

```
public java.lang.String getError()
```

Obtiene el valor del mensaje de error.

Returns:

devuelve el mensaje de error.

Package Class Tree Deprecated Index Help

Prev Class Next Class Frames No Frames All Classes

Summary: Nested | Field | Constr | Method Detail: Field | Constr | Method

Class Reloj

java.lang.Object
    Reloj

All Implemented Interfaces:

java.io.Serializable

```
public class Reloj
extends java.lang.Object
implements java.io.Serializable
```

Define un reloj.

Version:

1.0

Author:

Juan Luis Navarro Rey

See Also:

Serialized Form

Constructor Summary

Constructors

Constructor and Description
`Reloj()` Constructor de la clase.

Method Summary

Methods

Modifier and Type	Method and Description
`long`	`getTime()` Averigua la hora.
`void`	`setTime(long hora)` Modifica la hora.

Methods inherited from class java.lang.Object

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Constructor Detail

Reloj

```
public Reloj()
```

Constructor de la clase. Inicializa el reloj.

Method Detail

getTime

```
public long getTime()
```

Averigua la hora.

Returns:

Devuelve la hora en milisegundos.

setTime

```
public void setTime(long hora)
```

Modifica la hora.

Parameters:

`hora` - Nueva hora del reloj en milisegundos.

Package Class Tree Deprecated Index Help

Prev Class **Next Class** Frames No Frames All Classes

Summary: Nested | Field | Constr | Method Detail: Field | Constr | Method