

Práctica 1: creación de un FTP básico

Álvaro del Rosal Egidio
Programación en entornos distribuidos

26 de marzo de 2012

Índice

1. Enunciado	3
1.1. Funcionalidad mínima	3
1.2. Extensiones a la práctica	3
1.3. Entrega de la práctica	4
1.4. Evaluación de la práctica	4
2. Explicación del protocolo implementado	6
2.1. Add user	6
2.2. Bye	6
2.3. Can read	7
2.4. Can write	7
2.5. Delete	7
2.6. Delete user	8
2.7. Hello	8
2.8. Is file	9
2.9. Is folder	9
2.10. Is hidden	9
2.11. Last modified	10
2.12. Login	10
2.13. Look	10
2.14. New folder	11
2.15. Not admin write	11
2.16. Read	12
2.17. Size	12
2.18. Users	12
2.19. Write	13
3. Diagramas UML	14
3.1. Servidor	14
3.2. Cliente	15
3.3. Cliente (GUI)	16
4. Otra información	17
4.1. Descarga	17
4.2. Configuración	17
4.3. Usuarios	17
4.4. Otras librerías	17

1. Enunciado

1.1. Funcionalidad mínima

Deberá desarrollarse una aplicación cliente-servidor que tenga la funcionalidad básica de un FTP. El servidor estará ejecutándose en un equipo remoto, y permitirá ejecutar diversas acciones sobre el sistema de ficheros de la máquina en la que corre. El cliente se ejecutará en otra máquina y permitirá realizar al menos las siguientes acciones:

- Listar el contenido de cualquier directorio de la máquina remota.
- Mostrar información sobre los ficheros de la máquina remota, como la fecha de creación, tamaño, si es un fichero de sólo lectura, si es un fichero oculto...
- Descargar un fichero de la máquina remota.
- Subir un fichero a cualquier directorio de la máquina remota, siempre y cuando tengamos permisos de escritura en dicho directorio de la máquina remota.
- Borrar un fichero de la máquina remota.
- Crear un directorio en la máquina remota.
- Borrar un directorio vacío en la máquina remota.

Para poder ejecutar comandos sobre la máquina remota será necesario autenticarse a través de un usuario y un password. El usuario y el password deberán guardarse en un fichero de configuración en el servidor, que será leído por la aplicación que corre en el servidor cuando esta arranca. Podrá haber varios usuarios diferentes que puedan tener permiso para acceder a la máquina remota. Antes de poder realizar ninguna acción sobre la máquina remota, ésta deberá verificar que el cliente posee las credenciales adecuadas.

Deberán gestionarse de modo adecuado posibles fallos en la ejecución de los comandos enviados al servidor (por ejemplo, que se pide el listado de un directorio que no existe, que se desea borrar un archivo que no existe o sobre el cual no se tiene permiso de escritura, etc.).

1.2. Extensiones a la práctica

- Permitir borrar directorios aunque no estén vacíos en la máquina remota (hasta 0.5 puntos extra).
- Permitir que haya dos tipos diferentes de usuarios. Los usuarios *administrador* serán capaces de ejecutar todos los comandos sobre la máquina remota. Los usuarios normales sólo podrán ejecutar aquellos comandos que no resulten en modificaciones permanentes en el sistema de ficheros remoto, y podrán subir archivos a un único directorio especial que deberá indicarse en el archivo de configuración del servidor (hasta 2 puntos extra).
- Permitir cancelar la descarga y la carga de ficheros en cualquier momento desde el cliente (hasta 0.5 puntos extra).
- Permitir que los usuarios tipo *administrador* puedan remotamente crear nuevos usuarios, tanto normales como administradores, y eliminar usuarios (hasta 1 punto extra).

- Realizar logging de todas las acciones que suceden en el servidor, guardando información de qué usuario realizó la acción, cuando la realizó y que acción se realizó, y desde que máquina cliente se envió el comando. La ruta en la máquina remota del fichero de log deberá poder configurarse a través del archivo de configuración donde también se encuentran los nombres de usuario y sus password (hasta 1 punto extra).
- Desarrollar dos posibles interfaces para el cliente: una basada en un sistema de ventanas, y otra basada en una consola (hasta 2 puntos extra, a depender de la calidad de ambas interfaces).
- Usar un sistema de control de versiones desde el principio del desarrollo de la práctica (hasta 1 punto extra).
- Permitir ejecutar cualquier comando nativo de la máquina remota (por ejemplo, `dir/ls`, `netstat...`), y mostrar el resultado de su ejecución (hasta 2 puntos extra para el primero que lo implemente; hasta 1 punto extra para los demás).
- Implementar alguna solución que permita una comunicación encriptada entre el cliente y el servidor (hasta 2 puntos extra para el primero que lo implemente; hasta 1 punto extra para los demás).

1.3. Entrega de la práctica

Los alumnos deberán entregar junto con el código fuente de la aplicación (y cualquier otro artefacto que forme parte de ella) un informe documentando la aplicación donde al menos debe constar la siguiente información:

- Una descripción detallada del protocolo de aplicación que se ha diseñado e implementado para la realización de la práctica.
- Un diagrama UML recogiendo las clases/paquetes más relevantes de la aplicación.
- En caso de haberse empleado librerías de terceras partes, información sobre qué librerías se han empleado y con qué fin se han empleado.

El alumno podrá incluir en dicho informe cualquier otra información adicional que considere que ayuda a documentar su trabajo.

La entrega de la práctica se realizará en formato electrónico a través de e-mail. El informe de la práctica deberá entregarse impreso y en formato electrónico. La fecha límite para completar tanto la entrega como la defensa de la práctica es el 10 de Abril.

Deberá transcurrir al menos un período de cinco días entre la entrega de la práctica y su defensa.

1.4. Evaluación de la práctica

- Los alumnos que implementen la funcionalidad básica, y su aplicación funcione correctamente tendrán hasta un 6 si la aplicación cliente se basa en comandos de consola, y hasta un 7 si la aplicación cliente se basa en una interfaz gráfica. Si la funcionalidad básica de la aplicación no funciona correctamente, el alumno no podrá superar la asignatura.
- A la nota de la funcionalidad básica, se le sumará la nota obtenida en las extensiones implementadas (en caso de que se haya implementado alguna).

- La nota de la práctica tendrá que ser validada por una defensa oral individual delante del profesor. Si existe evidencia de que el alumno ha copiado, el alumno no superará la asignatura independientemente de la nota que la práctica se pudiese merecer. Se considerará evidencia de haber copiado el describir de modo incorrecto el funcionamiento de la práctica, o no comprender parte de la práctica.
- Las prácticas podrán desarrollarse individualmente o en parejas. En caso de realizar la práctica en pareja, la nota máxima (a depender de la defensa individual) que podrá obtener cada uno de los alumnos será igual a la nota de la práctica menos 3 puntos. En el caso de prácticas desarrolladas en pareja, ambos alumnos deberán de comprender y saber explicar perfectamente el funcionamiento de toda la práctica. Si alguno de los alumnos no entiende o no es capaz de explicar correctamente el funcionamiento de toda la práctica, no superará la asignatura.
- A lo largo de la asignatura se propondrán trabajos voluntarios que podrán sumar hasta 1 punto adicional a la nota conseguida en la práctica.

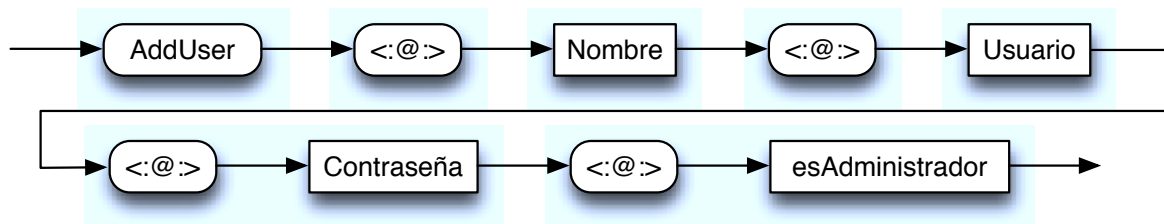
2. Explicación del protocolo implementado

El protocolo implementado está basado en FTP y permite los siguientes comandos o peticiones. La gramática de cada petición se expresa mediante *Diagramas de Conway* en los que los símbolos no terminales se expresan en rectángulos y los terminales mediante rectángulos redondeados. Los símbolos se relacionan mediante flechas que indican el flujo de interpretación.

2.1. Add user

Este comando permite agregar un nuevo usuario a la aplicación. Exige permisos de administrador.

Petición:



Respuesta:

(Sin respuesta) →

2.2. Bye

Este comando desconecta ordenadamente al usuario de la aplicación.

Petición:



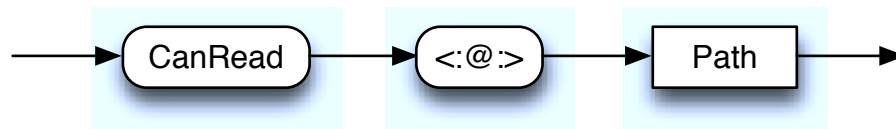
Respuesta:



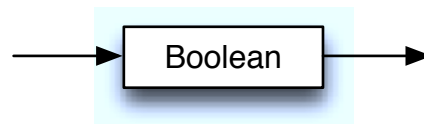
2.3. Can read

Este comando indica si un path es legible.

Petición:



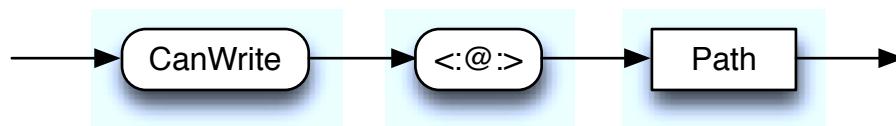
Respuesta:



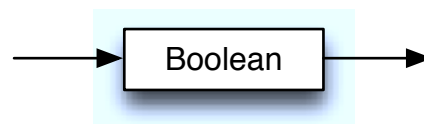
2.4. Can write

Este comando indica si un path se puede escribir.

Petición:



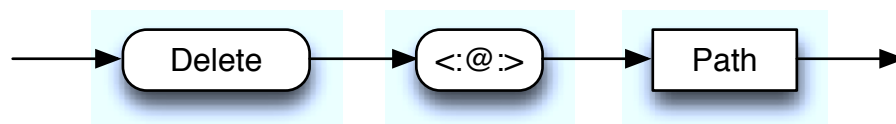
Respuesta:



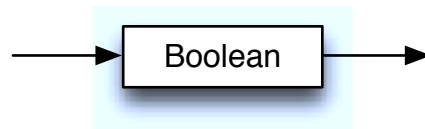
2.5. Delete

Este comando borra el path que se le indica. Si el path es un directorio y contiene archivos dentro, también los borra. La respuesta indica si se ha podido llevar a cabo correctamente la operación. Exige permisos de administrador.

Petición:

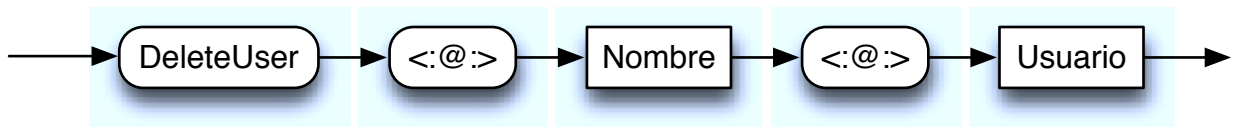


Respuesta:

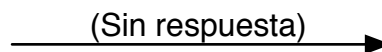


2.6. Delete user

Este comando borra el usuario que se le indica. Exige permisos de administrador.
Petición:

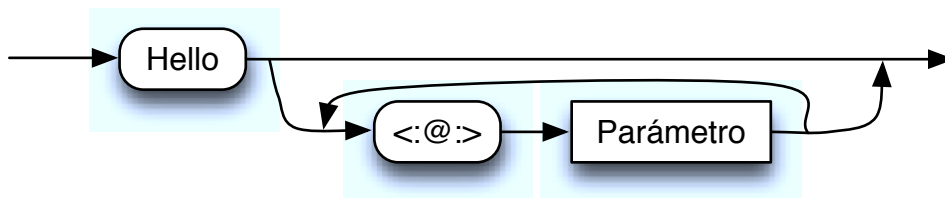


Respuesta:

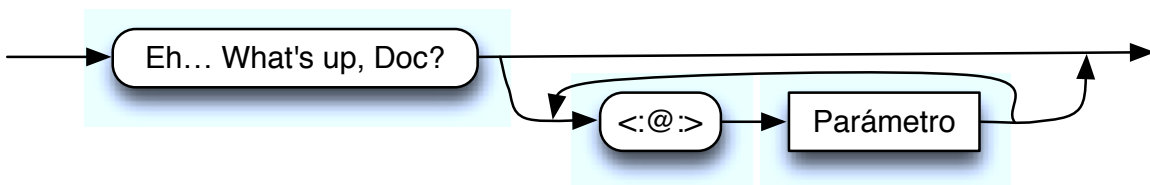


2.7. Hello

Este comando devuelve los mismos parámetros que se le pasan. Actualmente no se utiliza, pero se usa en pruebas.
Petición:



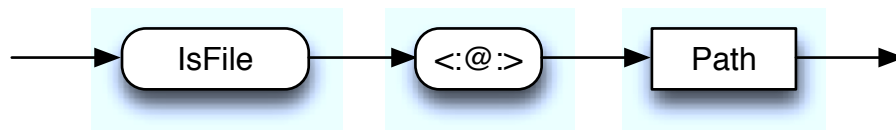
Respuesta:



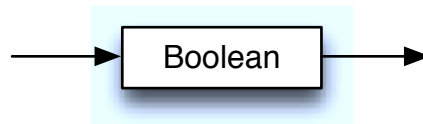
2.8. Is file

Este comando indica si el path indicado es un archivo.

Petición:



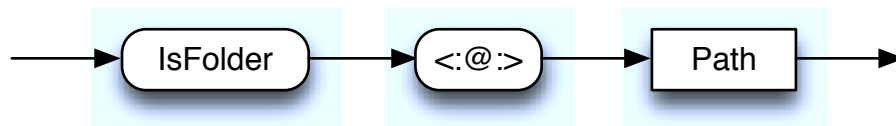
Respuesta:



2.9. Is folder

Este comando indica si el path indicado es una carpeta.

Petición:



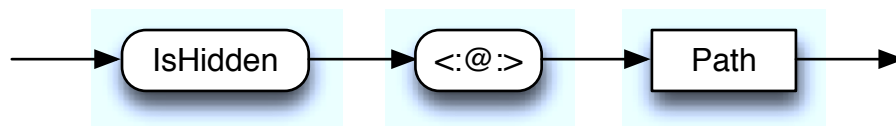
Respuesta:



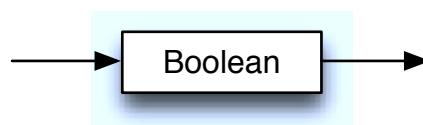
2.10. Is hidden

Este comando indica si el path indicado está oculto.

Petición:



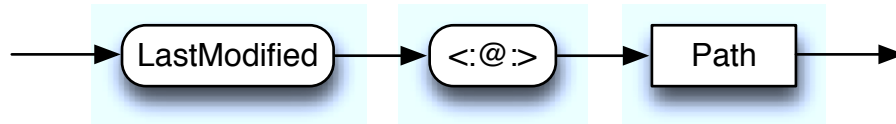
Respuesta:



2.11. Last modified

Este comando indica el número de milisegundos transcurridos entre las 00:00:00 del 1 de enero de 1970 y la última modificación del archivo.

Petición:



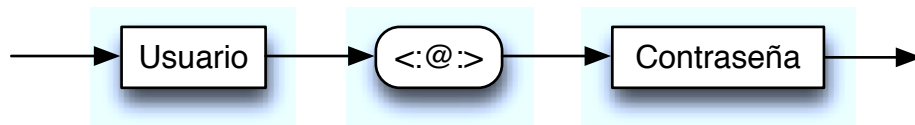
Respuesta:



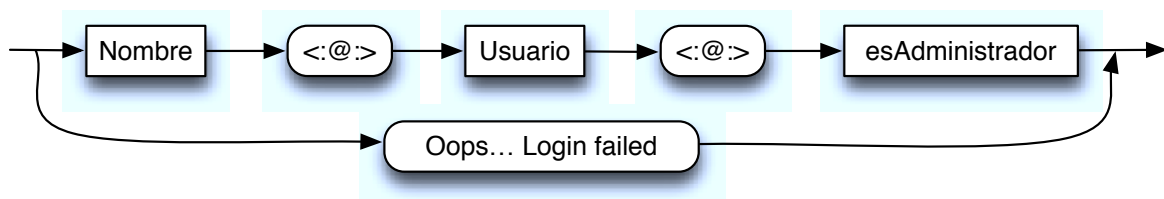
2.12. Login

Este comando es el primero en ser ejecutado. Inicia la sesión en el servidor y en caso de no ejecutarse, el servidor bloquea la ejecución de otros comandos hasta que se realice el login.

Petición:



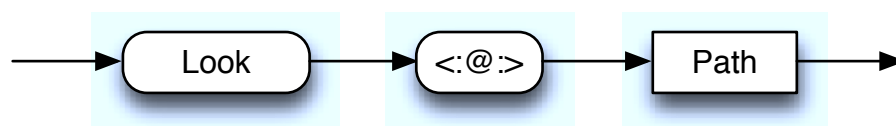
Respuesta:



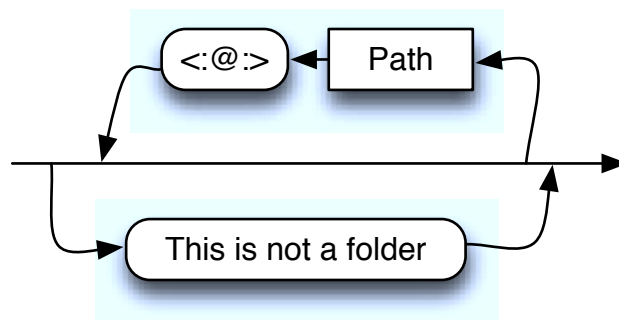
2.13. Look

Este comando devuelve los hijos de una carpeta. En caso de que esté vacía no devuelve nada y un mensaje de error si no es una carpeta.

Petición:



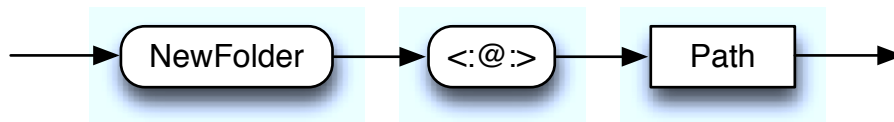
Respuesta:



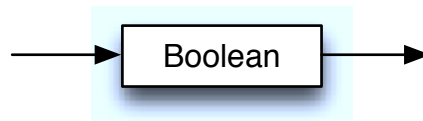
2.14. New folder

Crea una carpeta nueva en ese path. Devuelve el resultado de la ejecución. Exige permisos de administrador.

Petición:



Respuesta:



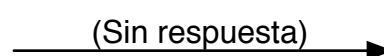
2.15. Not admin write

Este comando sube un archivo a la carpeta de entrada indicada en el archivo de configuración con el nombre indicado en el parámetro. En caso de que exista, el archivo se reemplaza. Los bytes del archivo están representados por enteros.

Petición:



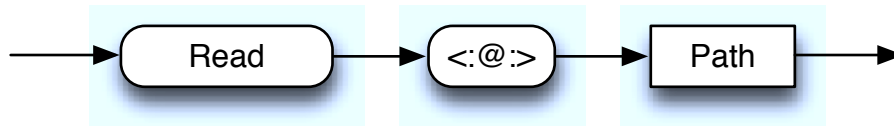
Respuesta:



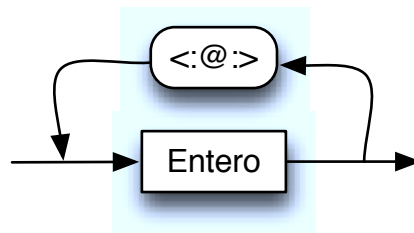
2.16. Read

Este comando envía el archivo leído al cliente que lo solicita. Los bytes están representados por enteros.

Petición:



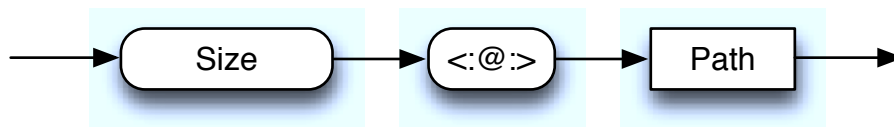
Respuesta:



2.17. Size

Este comando devuelve el tamaño en bytes del archivo. Si el path indicado es una carpeta, el valor es desconocido.

Petición:



Respuesta:



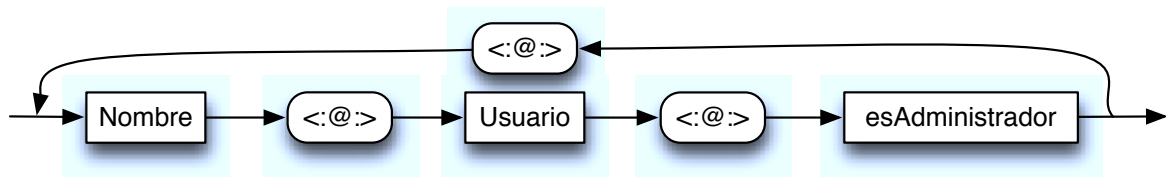
2.18. Users

Este comando devuelve todos los archivos registrados en el sistema. Exige permisos de administrador.

Petición:



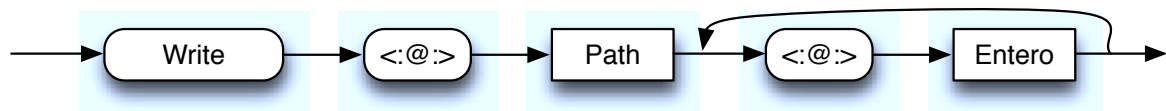
Respuesta:



2.19. Write

Este comando crea un archivo en el path indicado (y si existe lo reemplaza), escribiendo los bytes indicados como enteros. Exige permisos de administrador.

Petición:

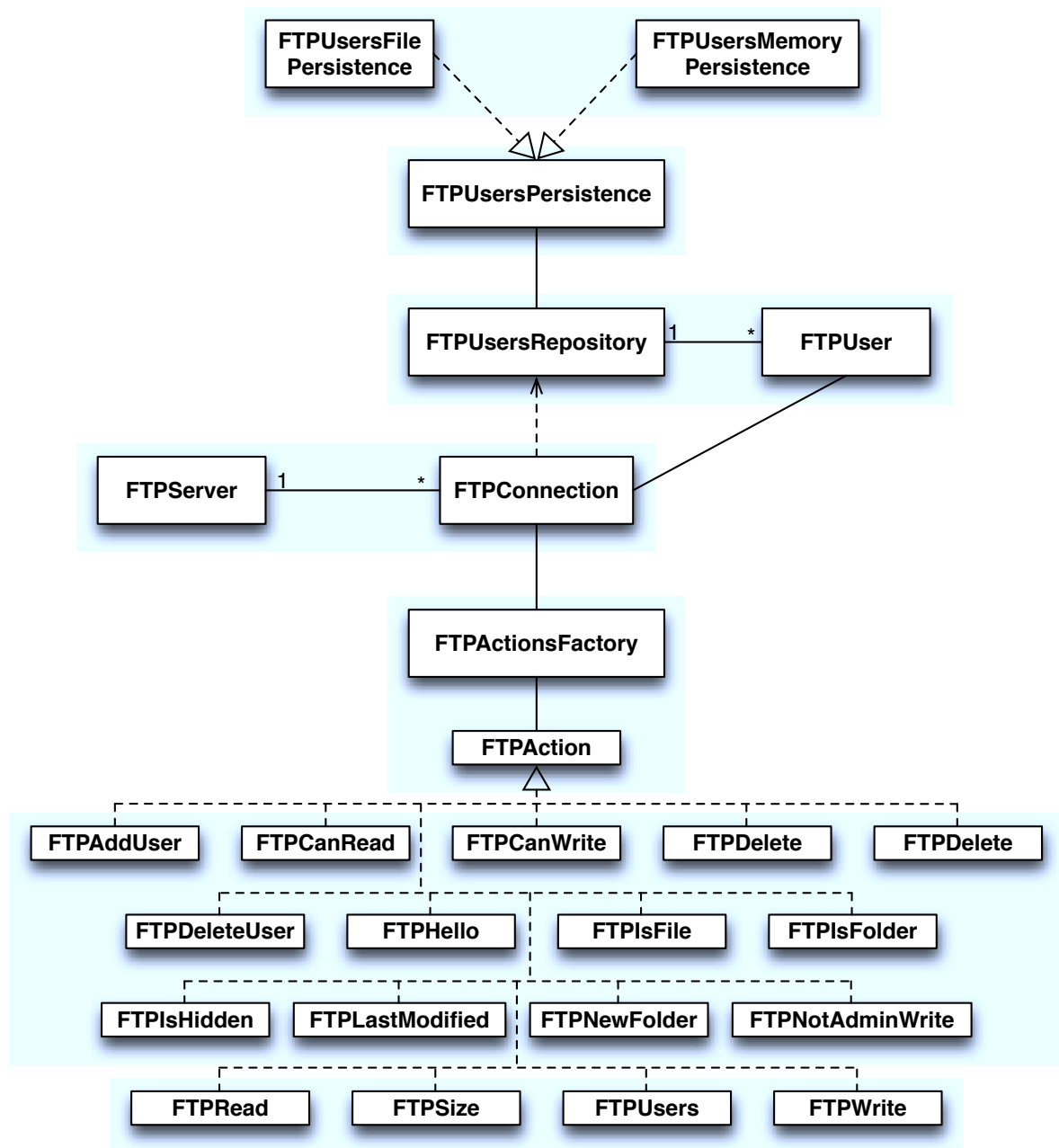


Respuesta:

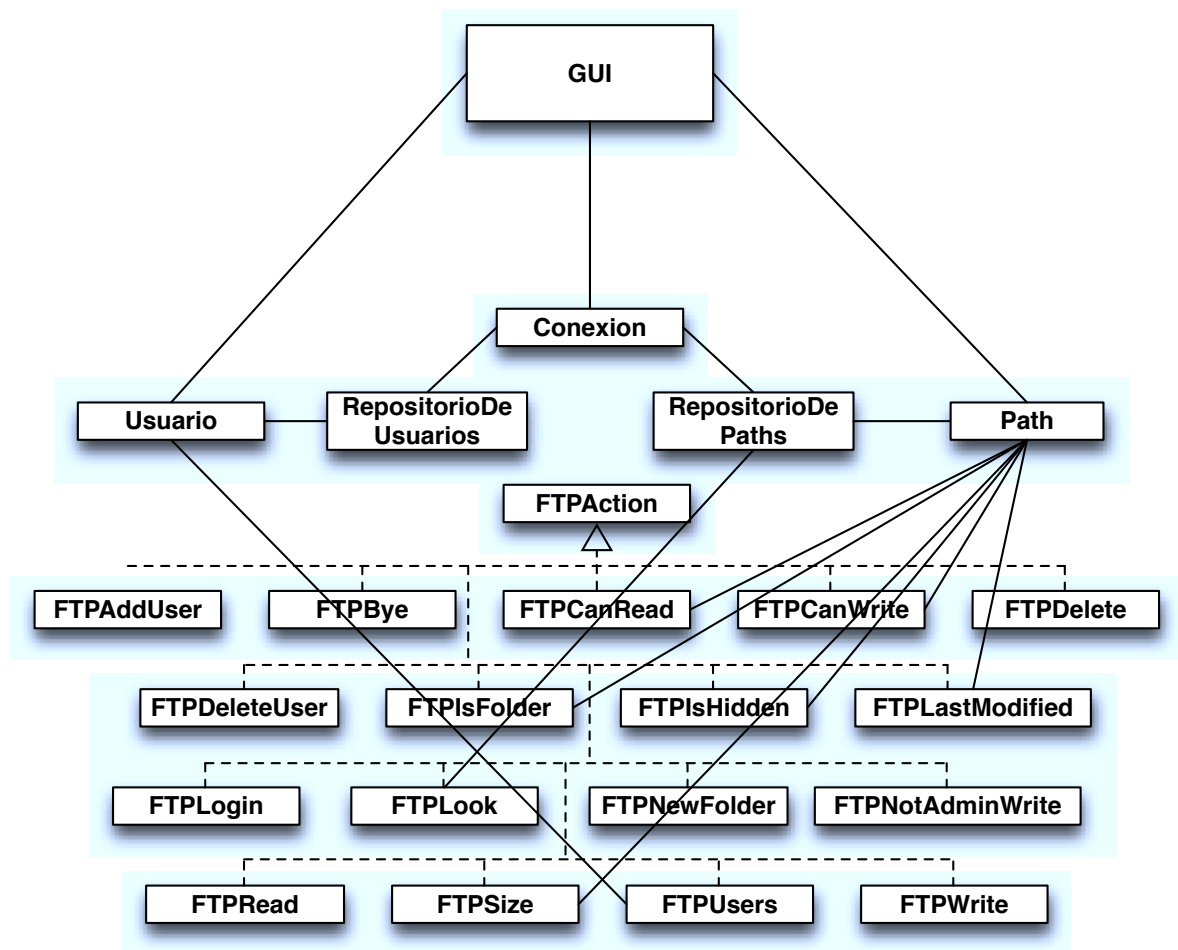
(Sin respuesta) →

3. Diagramas UML

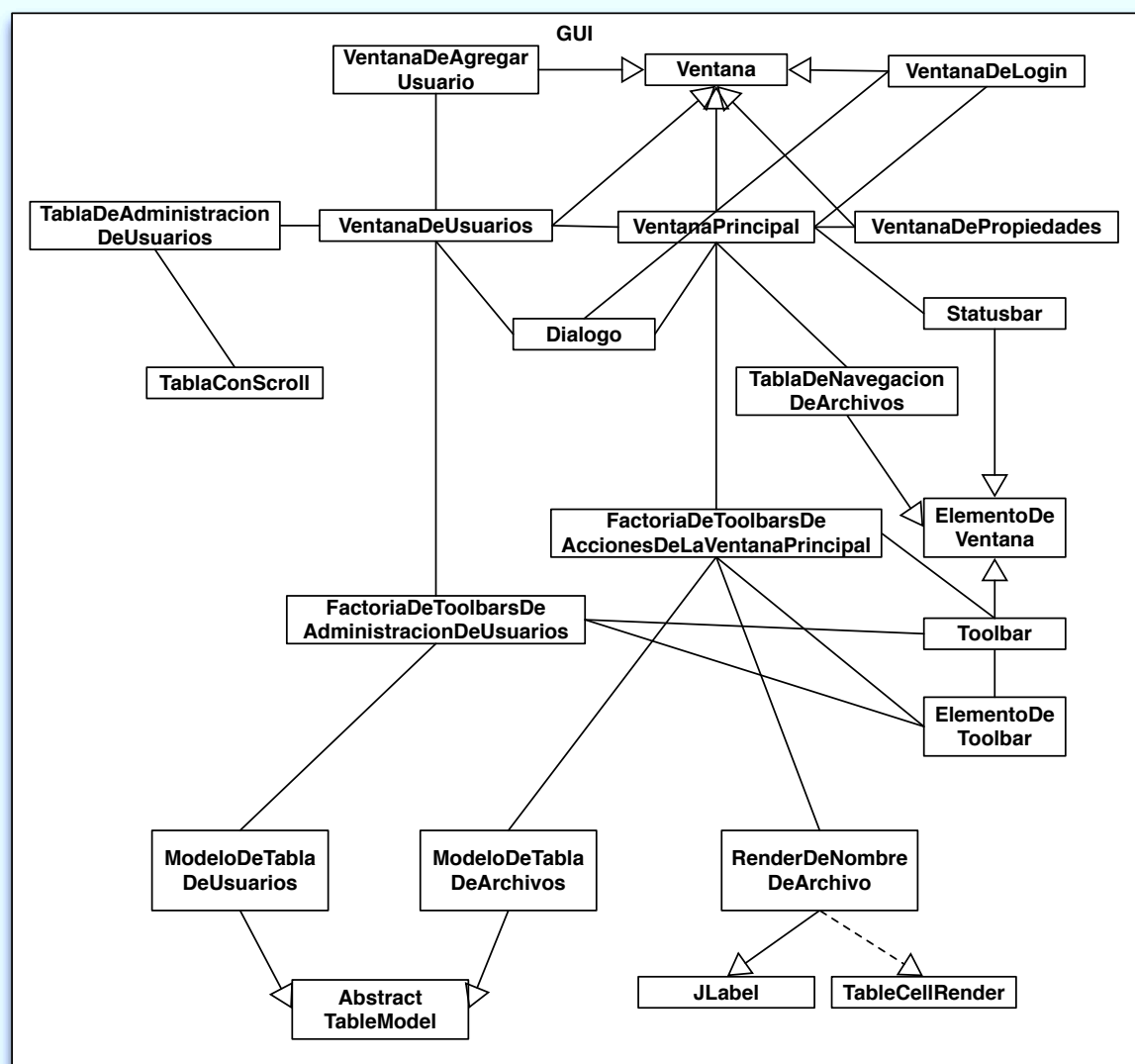
3.1. Servidor



3.2. Cliente



3.3. Cliente (GUI)



4. Otra información

4.1. Descarga

Este documento junto con un servidor y cliente que implementan una versión de este protocolo se encuentran en <http://code.google.com/p/pseudo-ftp/> (servidor) y <http://code.google.com/p/pseudo-ftp-client/> (cliente).

4.2. Configuración

En el mismo directorio que se encuentre el servidor, se debe localizar un archivo llamado `config.txt` como el que se muestra a continuación

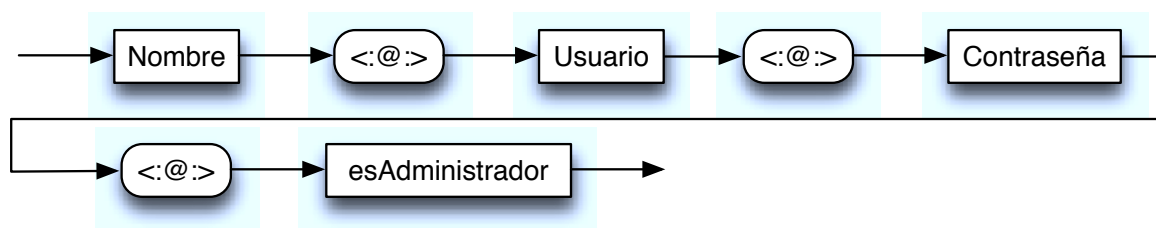
1. `users.txt`
2. `/Users/alvarodelrosal/inbox`

en la primera línea, debe incluirse la uri hasta el archivo donde se encuentren los usuarios registrados. Este archivo debe tener privilegios de lectura y escritura. En la segunda línea, se encuentra la carpeta de inbox donde los usuarios no administradores, tendrán permisos de subida de archivos; esta carpeta deberá tener privilegios de escritura y se deberá escribir sin la barra final.

4.3. Usuarios

Existen dos tipos de usuarios *administradores* y *usuarios*. Los *administradores* tienen acceso ilimitado y pueden crear carpetas, borrar carpetas y archivos, subir archivos a cualquier directorio del disco y administrar (crear, modificar y borrar) usuarios además de las funciones que pueden realizar los *usuarios*. Los *usuarios* solo pueden moverse con libertad por el sistema de archivos, subir documentos a un inbox, descargar cualquier archivo y ver las propiedades de cualquier elemento del sistema de archivos.

Estos usuarios se encuentran en el archivo definido en la segunda línea del archivo `config.txt`. El archivo incluye un usuario por línea, con una gramática como la siguiente:



donde `esAdministrador` puede tomar los valores `true` o `false` según sea administrador o no. A continuación se muestra un archivo de ejemplo:

1. `Alvaro<:@:>adr<:@:>123456<:@:>true`
2. `User<:@:>user<:@:>123456<:@:>false`
3. `Abraham<:@:>abraham<:@:>123456<:@:>true`

4.4. Otras librerías

En el desarrollo de la práctica se han empleado librerías de terceras partes como *Swing* para el desarrollo del GUI.