|  |
| --- |
| PROGRAMACIÓN DE SERVICIOS Y PROCESOS |
| Servicios en red |
| FTP y (SMTP/POP3) |
| **Manuel Fernandez**  **Jorge Zotano** |
| **Gonzalo Marinucci** |
| **16/01/2019** |

|  |
| --- |
| Manual de usuario sobre la utilización e instalación de los servicios FTP (Transmisión de archivos) y envío y recepción de correos electrónicos (SMTP/POP3) |

ÍNDICE

1. Capítulo 1. Transmisión de archivos (FTP)
   1. Introducción
   2. Requisitos
   3. Funcionamiento
   4. Implementación
2. Capítulo 2. Envío y recepción de correos electrónicos (SMTP/POP3)
   1. Introducción
   2. Requisitos
      1. Librerías necesarias
      2. Instalación
   3. Implementación
   4. Funcionamiento
3. **CAPÍTULO 1. TRANSMISIÓN DE ARCHIVOS (FTP).**

**1.1 INTRODUCCIÓN**

En este capítulo veremos que es un servidor FTP, como funciona y como instalarlo.

FTP es un protocolo de red para la transferencia de archivos entre sistemas conectados a una red TCP, basado en la arquitectura cliente-servidor.

Desde un equipo cliente se puede conectar a un servidor para descargar archivos desde él o para enviarles archivos, independientemente del sistema operativo utilizado en cada equipo.

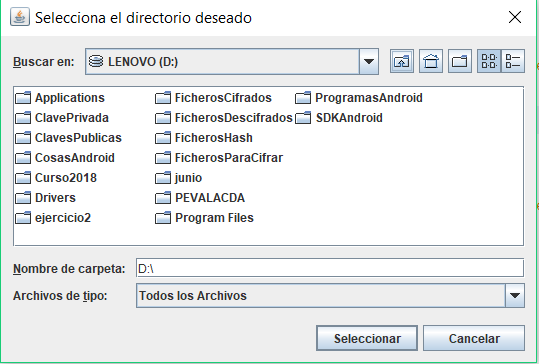
**1.2 REQUISITOS**

En este programa, gracias a que no utilizamos servidores dados por terceros, no es necesario ningún paquete complementario ni instalación. Esto es gracias a que nosotros mismos creamos un servidor y hacemos que el usuario se conecte a este sin necesidad de internet y similares.

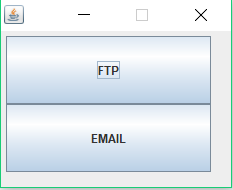
**1.3 FUNCIONAMIENTO**

En este apartado mostraremos una guía para el usuario de cómo usar el FTP.

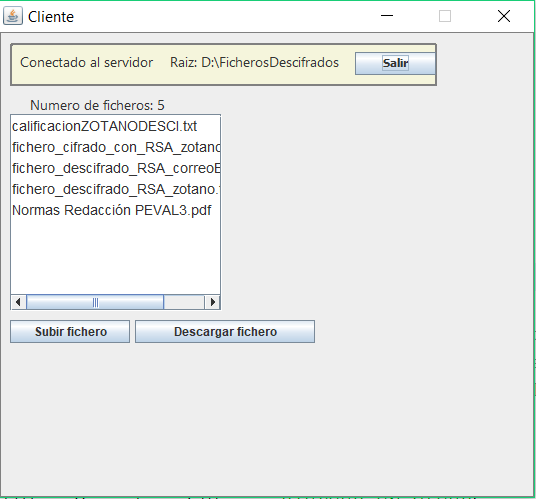
Lo primero que debemos hacer es ejecutar la clase “Servidor” para abrir nuestro servidor local. Inmediatamente después nos aparecerá una ventana en la cual deberemos elegir un directorio en el cual trabajar, ese directorio se usará como servidor y obtendremos los datos de ese sitio.



A continuación ejecutamos la clase “Principal” la cual abrirá una ventana con dos botones, pulsamos es FTP para acceder como cliente.

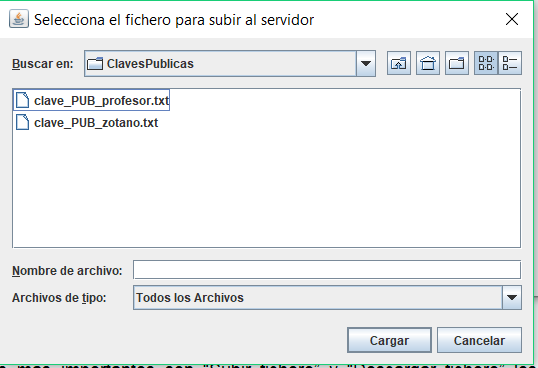


Después de entrar como cliente se abrirá una ventana con los datos del servidor y varios botones que realizarán distintas acciones.

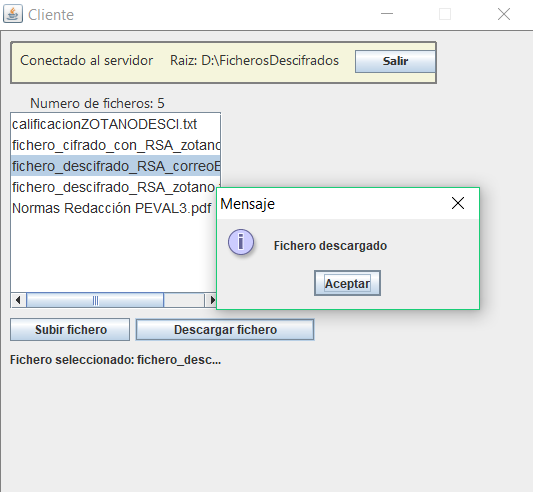


Los botones más importantes son “Subir fichero” y “Descargar fichero” los cuales nos permiten subir un fichero al servidor y descargar un fichero de este respectivamente.

Para subir un fichero pulsaremos dicho botón, el cual abrirá una ventana donde elegiremos el fichero a subir.



Para descargar un fichero del servidor pulsaremos el botón correspondiente después de haber seleccionado un fichero de la lista.

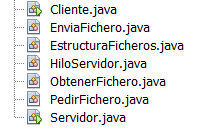


El fichero seleccionado se descargará directamente en la carpeta del proyecto, si se descarga un fichero y hay uno que se llama igual, este se sobrescribirá.

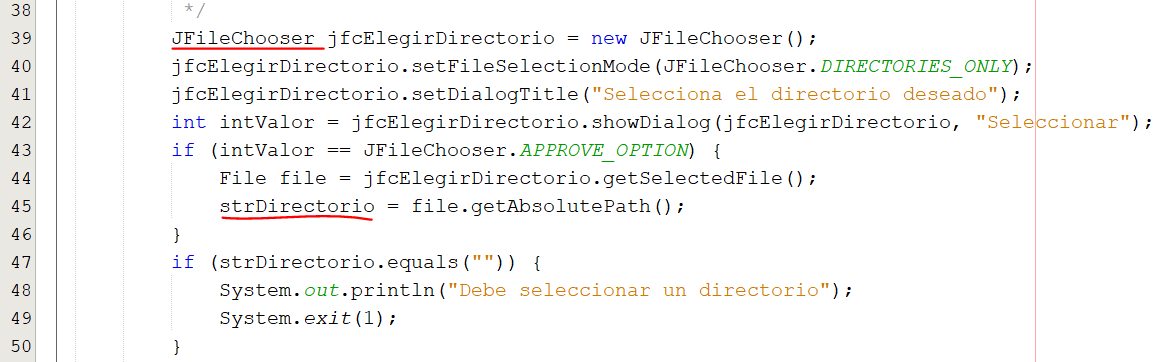
Para que el usuario se desconecte puede pulsar el botón “Salir” o pulsar la cruz en la parte superior derecha de la ventana.

**1.4 IMPLEMENTACIÓN**

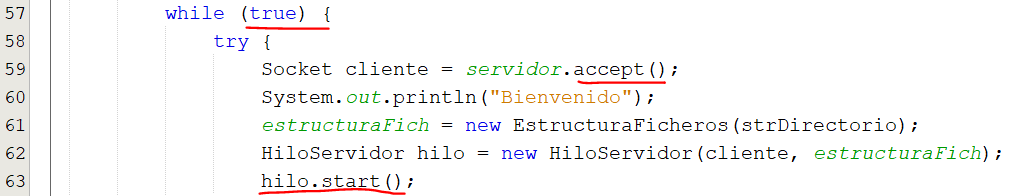
Este módulo se compone de siete clases diferentes, Servidor, Cliente, HiloServidor, EnviarFichero, EstructuraFichero, ObtenerFichero y PedirFichero.



La clase Servidor se encargará de abrir un servidor local en un directorio que el mismo cliente seleccionará.

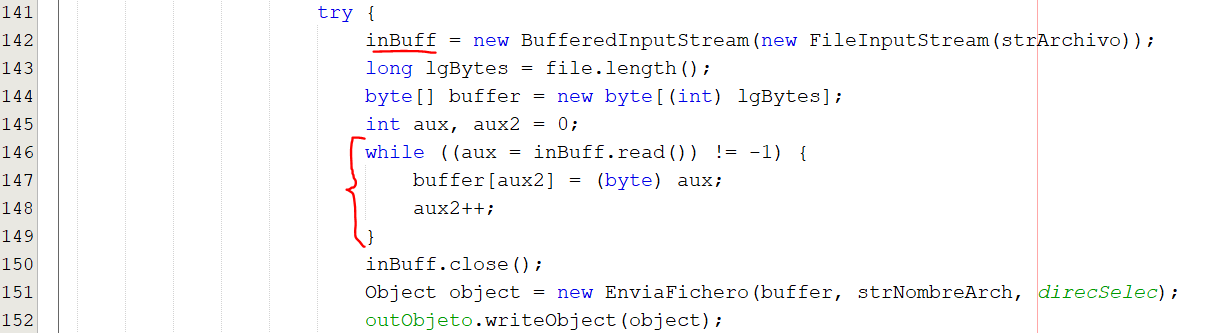


El servidor estará siempre en bucle para que se puedan conectar clientes siempre que quieran.

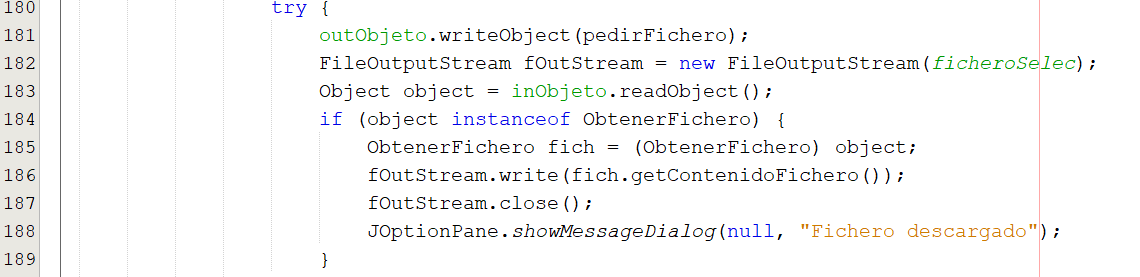


La clase cliente se basa en una ventana que mostrará los datos del servidor y las acciones que podemos realizar. Esta clase tiene varios métodos interesantes para comentar.

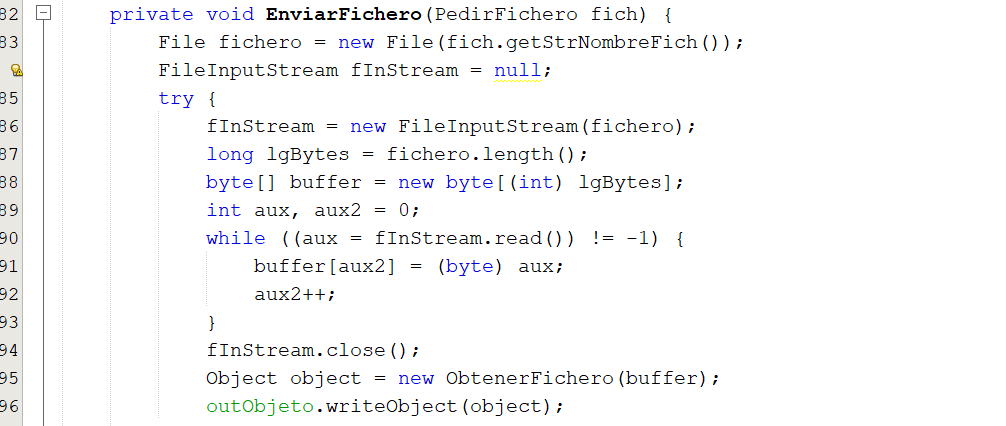
El primero sería a la hora de pulsar el botón para subir un fichero al servidor. Este método funciona gracias a que un buffer recoge el contenido del fichero seleccionado y escribe uno nuevo idéntico.



Otro evento se produciría cuando el cliente pulsara el botón de descargar fichero, cuando esto sucede se guarda en una variable objeto la estructura del mismo, la cual contiene la información del fichero, y se crea una copia en el directorio donde esté almacenada nuestra aplicación.



La clase HiloServidor será usada como una tubería para compartir datos del cliente y del servidor para llevar a cabo métodos, como por ejemplo el método EnviarFichero el cual necesita la ruta del servidor y la ruta del fichero que el cliente quiere subir al servidor.



Las clases PedirFichero, ObtenerFichero y EnviarFichero se usan como objetos que se usarán dependiendo de qué acción quiera llevar a cabo el cliente. La clase EstructuraFichero es la que más información almacena ya que guarda todo lo relacionado con el servidor tal como la ruta, nombre del directorio y un array con todos los datos que hay dentro del servidor.

1. **CAPÍTULO 2. ENVÍO Y RECEPCIÓN DE CORREOS ELECTRÓNICOS (SMTP/POP3).**

**2.1 INTRODUCCIÓN**

En este capítulo veremos que es un servidor SMTP, como funciona y como instalarlo, también usaremos el POP3.

El SMTP es el protocolo estándar de internet para el intercambio de correo electrónico. Funciona con comandos de texto que se envían al servidor. Cada vez que se envía un comando el servidor SMTP envía una respuesta.

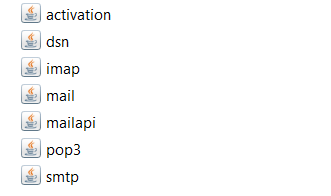
La pega de crear un servidor SMTP es que solamente nos proporciona la opción de mandar correos electrónicos por lo tanto debemos apoyarnos del POP3 el cual nos permite ver los correos electrónicos existentes en el servidor.

**2.2 REQUISITOS**

A diferencia del módulo anterior esta vez tendremos que usar varias librerías para hacer que nuestro programa funcione.

**2.2.1 Librerías necesarias**

Las librerías que necesitaremos son las siguientes, activation, dsn, imap, mail, mailapi, pop3 y smtp.



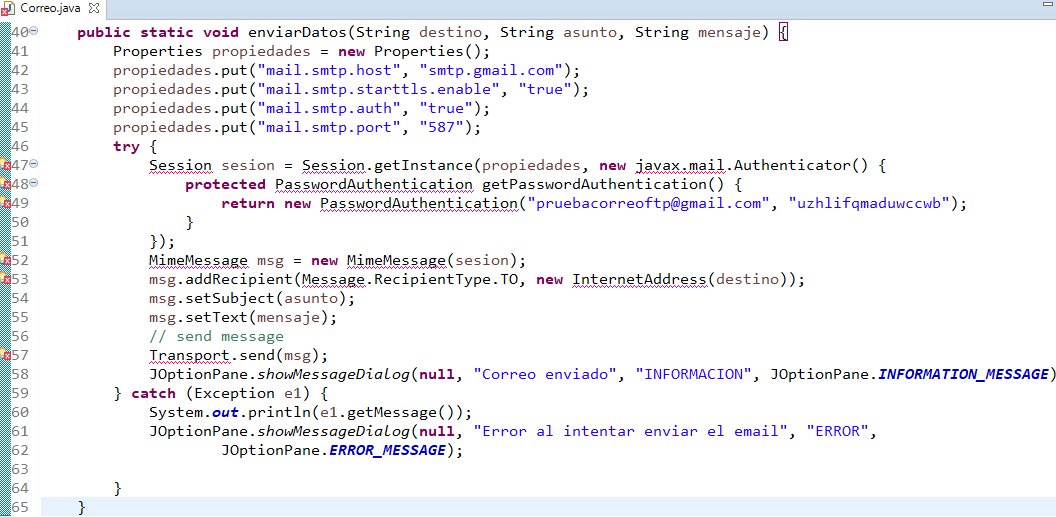
**2.2.2 Instalación**

En este caso no necesitamos instalar nada ya que como se indica en la peval las librerías deben estar ya importadas, en el caso contrario debemos descargar los .jar desde sus respectivas páginas e importarlos.

**2.3 IMPLEMENTACIÓN**

El módulo de email se compone de la siguientes clases

La clase Correo contiene un método enviarDatos, que es la encargada de enviar datos acerca del correo como el destino, asunto y mensaje del correo

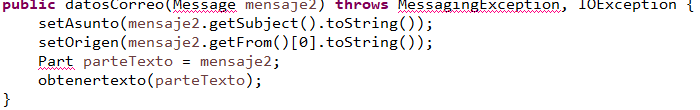


También contiene un método recibirDatos, cuya funcionalidad es la de recibir correos.

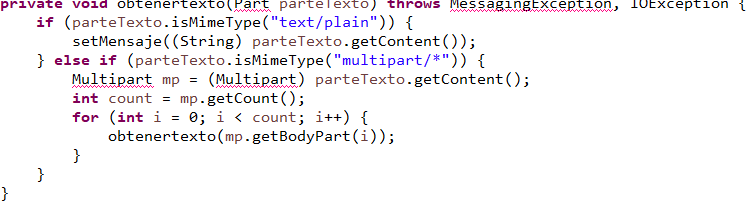
Obtendrá del almacen de la carpeta INBOX los mensajes que contenga y los almacenará en el array de Message



La clase datosCorreo tiene un **constructor** recibe como parámetro un objeto Message donde le indicamos que el asunto y el origen los recibirá de mensaje2 y se los establecemos mediante los métodos setters de la clase Message



También contiene un **método** obtenerTexto el cual compara si el texto es plano o de tipo multiparte, si es de tipo texto plano llama a set mensaje para pasarle como parámetro el contenido de texto del texto del parámetro



**2.4 FUNCIONAMIENTO**