

12-6-2020

FTP - Practice

Networks & Communications II



universidad
SANJORGE
GRUPO SANVALERO

Adriana Bastera
Lin Hirwa Shema
Juan Ormaechea

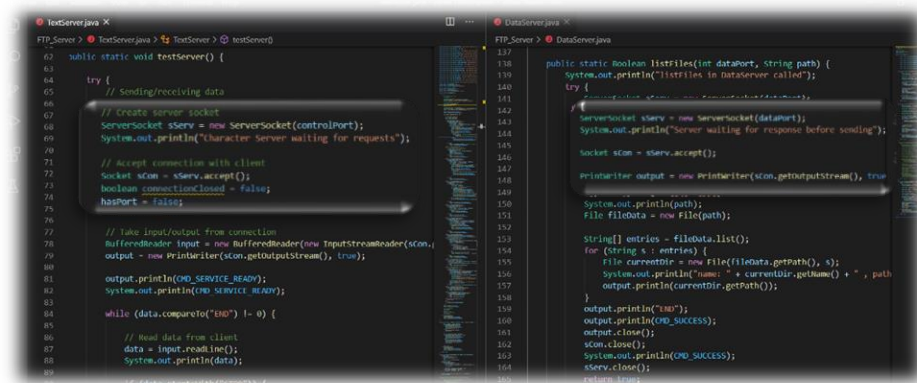
INDEX

1. Introduction	page 1
2. Basic Sequence	page 1
3. Optional Modules	page 1
I. Authentication	page 2
II. Directory system	page 2
III. Multithread server	page 2
IV. Graphic user interface	page 3
V. Delete & Rename files	page 4
4. Conclusions	page 5

Introduction

The development of this practice consists of implementing the FTP protocol using the TCP transport protocol. To do this, both the server and the client have had to be simulated in order to make file transfer possible.

In addition, since FTP is an application layer protocol for file transfer, two connections have been required: the first dedicated to command-based client-server communication and the second focused on data and file transmission.



Different connections of classes TextServer & DataServer

Basic sequence

In the basic sequence the objectives were defined and clear of which the indicated ones have been done:

- | | |
|--|--|
| - Active mode connection | ☑ Active mode connection |
| - Single directory | ☑ Single directory |
| - List files in directory | ☑ List files in directory |
| - Receive files | ☑ Receive files |
| - Download files | ☑ Download files |
| - Binary transference mode (files) & ASCII (list of files) | ☑ Binary transference mode (files) & ASCII (list of files) |
| - Configurable port | ☑ Configurable port |
| - Hidden FTP commands in client side | ☑ Hidden FTP commands in client side |
| - Simple usage | ☑ Simple usage |

Optional modules

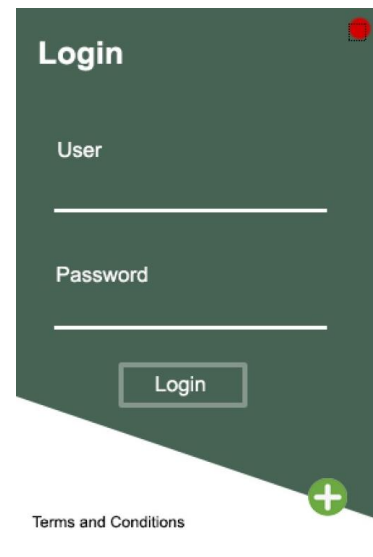
Among the optional modules, we have discarded the user control, the multithread server, and the passive mode modules, and proceeded to develop the following ones:

1. Authentication system

For the authentication system, as soon as you start the connection with the client, you are asked for the unique username and password, being:

User: <u>user</u>	Password: <u>password</u>
-------------------	---------------------------

This prevents any user from accessing the files on the server side. Once logged in and the user and password have been checked, the client is given the opportunity to download, rename, delete, or upload files.



2. Directory system

The only objective of this section is to allow the user to create different directories within the server in order to classify the files and be able to find them more easily. This makes the project more complex, since the user must be able to create, delete and rename directories and have access to the files contained in them in the same way.

3. Multithread server

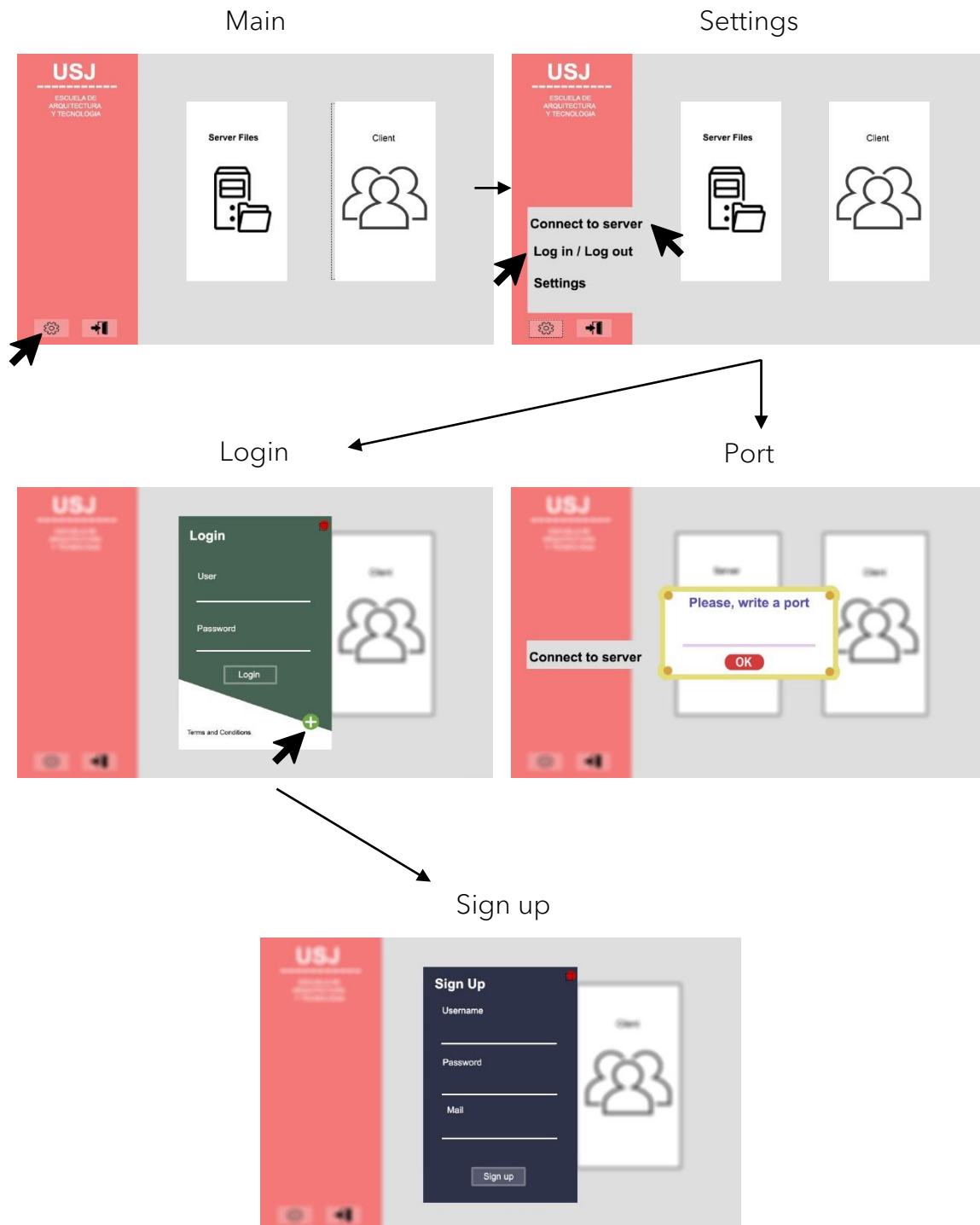
The multithread server aims to be able to deal with different clients at the same time, responding to their requests simultaneously.

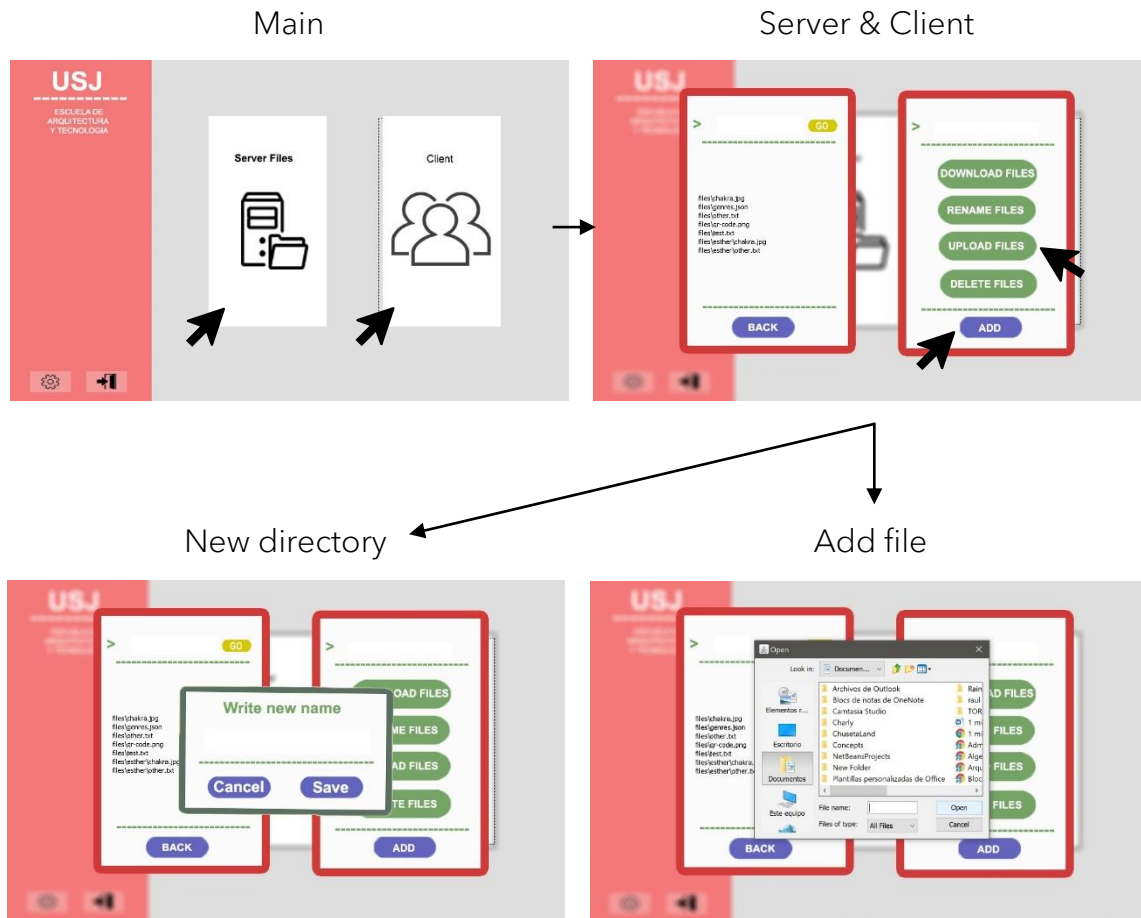
The problem is that if a client changes the working directory, it is changed in all of them. They all share the same folder and also can login with the same username and password, but they are usually differentiated by the port number. Also, when the client enters or exits, a coloured message appears in the console. If it enters green and if it leaves red.

4. Graphic user interface for the client

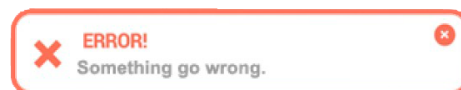
This optional module makes it easier for the user to locate the files and shows them more clearly with abstraction so that the experience is more pleasant.

This interface is composed of x windows, each with different functionalities but which, connected to each other, meet the objectives described above.





For the rest of the actions, simply press the corresponding buttons to perform the actions indicated by the button name itself. Every time an action is performed; a message is displayed:



5. Delete and rename files

As the name of this module indicates, the only goal is to be able to rename and delete files from any directory regardless of the directory where the client is currently located. Also, it allows the client to rename and delete directories from inside the working directory.

Conclusions

Thanks to the development of this practice, it has been possible to understand at first hand the implementation of protocols at application level, as well as their utilities, taking into account that this protocol is widely used by many services on the Internet nowadays.

But beyond the protocol and its implications, this practice has helped to improve the programming skills of all the team members, in addition to learning how to manage, create and connect the different parts of the interface, handle the clients' users and passwords and have more fluency with the version controls so necessary in team works.