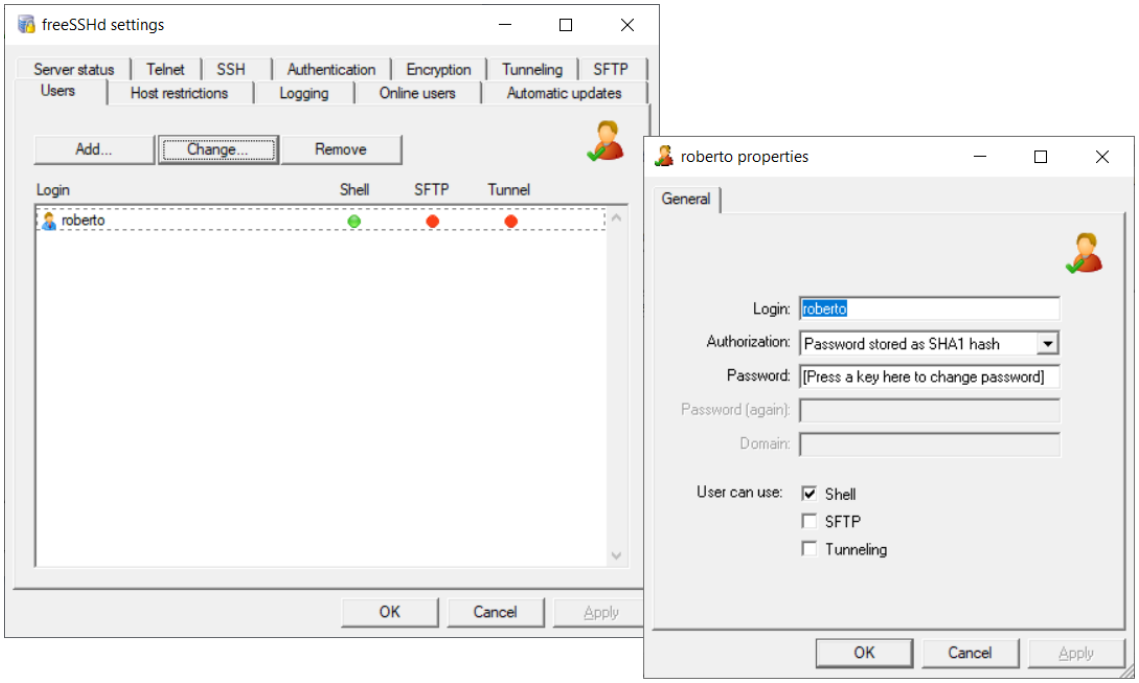
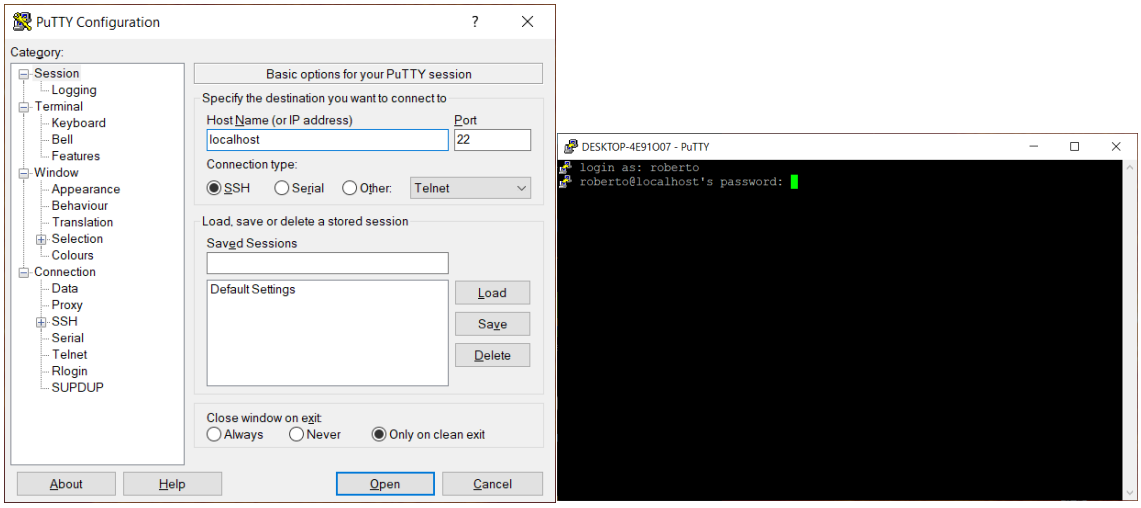
Cuaderno de ejercicios

Tema 5 - Generación de servicios en red

1. Descarga e instala el servidor SSH FreeSSHd y el cliente Putty. En el FreeSSHd crea un usuario con autorización por contraseña y que pueda usar el shell:



Este usuario y contraseña es el que deberás utilizar para conectarte con el cliente ssh Putty.



Realiza pruebas en localhost y también conectándote al PC de un compañero. Prueba algunos comandos en la consola, como explorar el contenido del disco, lanzar alguna aplicación, crear un directorio, etc.

2. Partiendo de lo visto en teoría, desarrolla un cliente de correo lo más sencillo que puedas. Para ello, primero crea y configura de forma adecuada una cuenta de correo de prueba en Gmail. Utilízala como remitente para enviar un mensaje sencillo a otra cuenta. La cuenta

destino puede ser de cualquier domino (prueba por ejemplo con la que tienes en @floridauniversitaria.es).

3. Amplía el cliente de correo para que admita también el envío de uno o más anexos en el correo, y además que se pueda enviar a varios destinatarios. Prueba de enviar el correo a varios compañeros a la vez.
4. (OPCIONAL) En la siguiente web se indican los pasos para crear un cliente de correo IMAP que permita descargar los mensajes del servidor (texto y anexos):

   https://www.codejava.net/java-ee/javamail/receive-e-mail-messages-from-a-pop3-imap-server

   Este código funciona bien cuando el contenido del mensaje es texto plano. Sin embargo, cuando el contenido está formado de varias partes (es decir, multipart, como el que tienes en el cliente que has implementado en ejercicios anteriores) hay que explorar cada parte del mensaje para ir sacando el contenido de texto de cada una. En este post se plantea una forma de hacerlo:

   https://stackoverflow.com/questions/11240368/how-to-read-text-inside-body-of-mail-using-javax-mail/31877854

   Intenta integrar la información de estas dos páginas para implementar un cliente de correo que descargue y muestre los mensajes de tu buzón INBOX (o de cualquier otro buzón de tu cuenta).

   A continuación puedes añadir también el código para descargar en alguna ruta local los anexos. Para ello te puedes basar en el siguiente post:

   https://www.baeldung.com/java-download-email-attachments

5. Implementa la clase ServidorHTTP vista en la teoría y revisa las instrucciones para ver qué hace cada una. Debes tener claro cuál es el papel del GestorHTTP y la ruta (contexto) desde el cual va a dar respuesta el servidor.
6. Implementa la clase GestorHTTP para que acepte peticiones de tipo GET. Programa un método que cuando llegue una petición GET muestre por la consola del servidor el contenido de la petición.
7. Continúa con la gestión de peticiones GET. Programa ahora un método que devuelva una página web (como String) que se mostrará en el navegador. Haz que se sirvan páginas web distintas en función de lo que el usuario indique en la URL.
8. Descarga e instala el programa Postman. Revisa por encima la documentación del programa y cómo funciona.
9. Amplía el GestorHTTP para que gestione también peticiones de tipo POST. Programa un método que cuando llegue una petición POST recoja línea a línea todo su contenido y lo muestre por la consola del servidor.
10. Realiza diferentes envíos (strings variados) desde Postman y comprueba que el servidor los recoge correctamente.
11. (EJERCICIO TEÓRICO-PRÁCTICO) Piensa qué tipo de objetos podrías enviar o recibir desde el servidor. ¿Podrían ser JSONs? Prueba mandar algún JSON como POST y que el GestorHTTP lo reciba y lo procese correctamente. ¿Cómo harías para transformar en un objeto Java un JSON que te llega como String a través de POST?
12. (EJERCICIO TEÓRICO) Piensa que tienes conectada al servidor una base de datos (por ejemplo, MySQL o MongoDB). ¿Podrías traducir las peticiones GET y POST en operaciones CRUD de sobre la base de datos? ¿Qué operaciones podrías hacer?
13. (EJERCICIO TEÓRICO) Hasta ahora las peticiones GET y POST las has realizado desde el navegador o desde Postman, respectivamente. ¿Has visto en Javascript algunas tecnologías

que te permitan realizar peticiones asíncronas de este tipo? ¿Crees que podrías utilizarlas para que una aplicación móvil se conectara a tu propio servidor para consumir y guardar datos? Piensa cómo lo harías y si las podrías utilizar en un proyecto.