Ejercicios Unidad 3

Programación en Red

1 Conexión con URL HTTP

Programa que reciba a través de un parámetro de línea de comando una URL de una página web y descargue todas las imágenes referenciadas en el parámetro src de cada etiqueta <img> de dicha página.

2 Servidor de la hora

Desarrolla un servidor que envíe la hora a sus clientes. Se trata de un servidor muy simple que lo único que hará será enviar la hora inmediatamente después de aceptar cada conexión y a continuación finalizarla sin más.

Desarrolla un programa cliente que haga una única petición al servidor cada vez que se ejecute y muestre la respuesta intentando corregir la hora recibida teniendo en cuenta el posible retraso en la recepción.

3 Servidor eco

Desarrolla con Java un sistema cliente/servidor en el que el servidor lea secuencias de caracteres enviadas por el cliente y se las reenvíe inmediatamente de vuelta. Ambas aplicaciones usarán la consola como dispositivo estándar de E/S.

4 Servidor aritmético

Desarrolla con Java un sistema cliente/servidor en el que el servidor acepte peticiones del cliente para realizar sumas, restas, multiplicaciones, divisiones y raíces cuadradas. Sólo una de estas operaciones por cada petición.

El cliente proporcionará una interfaz gráfica de usuario basada en Swing con el aspecto que tradicionalmente presenta cualquier calculadora, además de incluir los controles necesarios para conectarse con el servidor, usando tanto una IP como un nombre de dominio.

5 Tres en raya

Desarrollar en Java un servidor del juego “tres en raya” permita el desarrollo de este juego entre múltiples parejas de clientes. Cuando un cliente se conecta, esperará a que un segundo cliente se conecte también, momento en el que comenzará el juego. El juego finalizará cuando uno de los dos jugadores gane o cuando alguno de ellos se rinda (finaliza la conexión antes de que concluya el juego).

Diseña el protocolo y crea el programa cliente con una interfaz gráfica de usuario basada en Swing.

6 Servidor de contactos

Crea un servidor que resuelva de forma concurrente peticiones de clientes para el manejo de una lista de contactos telefónicos, teniendo en cuenta que cada contacto puede tener varios teléfonos.

No se pondrá límite al número de contactos que se pueden almacenar en el servidor ni a la cantidad de teléfonos que puede tener un contacto, salvo el que impone el tamaño de la memoria disponible.

Los clientes realizarán una petición por conexión en formato de texto, atendiendo a la sintaxis que se describe a continuación:

Añadir un contacto:

```
nombre:teléfono
```

- El dato *teléfono* estará formado únicamente por dígitos decimales.
- Si se especifica un nombre nuevo, se crea un nuevo contacto con el teléfono especificado.
- Si el contacto ya existe, se le añade el teléfono especificado.
- El servidor retornará al cliente una línea con el texto "OK" si los datos se añaden correctamente.
- Si el teléfono ya existe para ese contacto, el servidor retornará una línea con el texto "ERR01".

Consultar los teléfonos de un contacto:

```
buscar:nombre
```

- El servidor retornará una línea con el texto "OK" seguida de los teléfonos del contacto, cada uno en una línea de texto.
- Si el contacto no existe, retornará una línea con el texto "ERR02".

Eliminar un contacto a partir de un nombre:

```
eliminar:nombre
```

- Si el contacto existe, se eliminan sus datos del servidor y éste retornará una línea con el texto "OK".
- Si el contacto no existe, retornará una línea con el texto "ERR02".

Mostrar todos los contactos:

```
contactos
```

- El servidor retornará una línea con el texto "OK" seguida de un listado con todos los contactos, cada uno en una línea de texto, en orden alfabético.

Si el servidor detecta un error de sintaxis en la petición, retornará al cliente una línea con el texto "ERR03", seguida de una línea con el texto de la petición, seguida de una línea que incluya el carácter '^' marcando la posición del error.

El cliente presentará la siguiente interfaz gráfica de usuario: