MEMORIA DE LA PRÁCTICA

“CONTROL DE ASISTENCIA”

Alejandro Díaz Obregón
PSP - 2ºDAM
2019/2020

KeyWords

Cliente, servidor, socket, hilo, ip, programa, ejecutar, asistencia, registro.

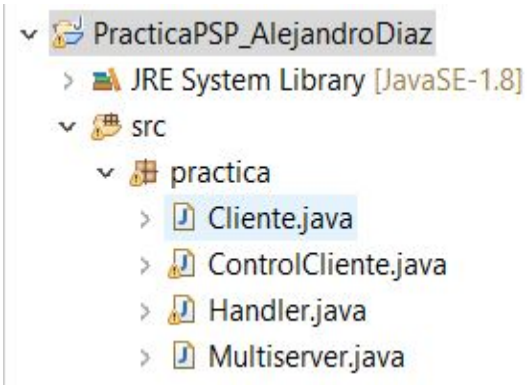
Resumen Proyecto

La práctica consiste en el desarrollo de un programa para controlar la asistencia de los alumno. Programa escrito en java.
Las necesidades de las que partimos son las siguientes:
→ Mecanismo para que los alumnos registren su asistencia.
→ Controlar que cada alumno no pueda registrar la asistencia de los demás, es decir, implementar un proceso de identificación único para cada alumno.
→ Controlar para qué clase se está registrando la asistencia.

Estructura

Basado en la estructura cliente-servidor.
Para la comunicación entre servidor y cliente se usarán sockets.
La parte de cliente corresponde al alumno.
La parte del servidor corresponde al profesor.
Como en cada clase habrá un profesor y varios alumnos, se implementará un *multiserver* para poder atender a las llamadas de los distintos clientes; por tanto se usarán *hilos* para las llamadas entre el cliente y el servidor.

En la siguiente imagen aparecen las clases creadas para el proyecto.



- ControlCliente.java
  El alumno deberá ejecutar esa clase.
  Lanza un hilo de la clase Cliente.java

- Cliente.java
  Contiene el código del programa para la parte del cliente.
  Crea un socket para comunicarse con el servidor.

- Multiserver.java:
  El profesor deberá ejecutar esta clase.
  Almacena en memoria los datos necesarios cuando un alumno registra una asistencia.
  Lanza un hilo de la clase Handler.java

- Handler.java:
  Contiene el código del programa para la parte del cliente.
  Recibe un socket creado por Multiserver.java para comunicarse con el cliente.

Cómo afrontar el problema

La asistencia se va a registrar en un fichero txt. Los datos que se almacenan en este fichero son **la ip y la hora** en la que cada cliente ha registrado la asistencia.

Para diferenciar la asistencia de cada clase se le pide al profesor que introduzca por teclado el nombre de su asignatura.

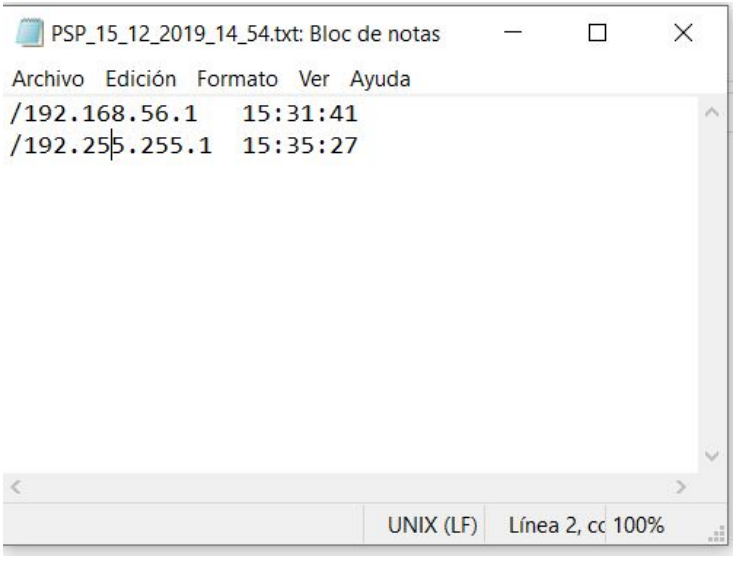
El fichero donde se registra la asistencia llevará el nombre de la asignatura seguido de la fecha y hora de creación de ese archivo, con el formato:

"Asignatura_dia_mes_año_hora_minutos.txt"

Ej: "PSP_15_12_2019_14_54.txt"

De esta manera cada vez que comience una clase, el profesor correspondiente ejecuta el servidor, y las asistencias de su hora se almacenan en un archivo con nombre identificativo único.

Un ejemplo del contenido del fichero sería el siguiente:



Resumen Funcionamiento

Antes de nada, el profesor debe ejecutar Multiserver.java para levantar el servidor y que éste pueda escuchar a los clientes.

Cuando un alumno ejecuta el programa, se le manda por parte del servidor una serie de claves numéricas creadas de forma aleatoria y una posición (también aleatoria) que designa la clave que el alumno deberá regresar al servidor.

Handler.java:

Cuando se recoge la clave introducida por el alumno, se comprueba que coincida con la correspondiente a la posición que ha generado el servidor. Si la clave es la correcta, se almacena en un HashMap la ip del socket del cliente y la clave numérica. Aparte se almacena en un ArrayList la hora a la que se ha registrado el cliente.

Hay que comprobar antes que no se haya registrado ya esa ip (comprobar que no se encuentra la ip en el HashMap).

Si todas las comprobaciones son correctas se vuelcan las ip con sus horas registro en el fichero creado específicamente para esa clase.

Multiserver.java seguirá funcionando mientras el profesor no cierre el programa, para que los alumnos puedan seguir registrándose durante la hora de clase.

En este caso no se han controlado los retrasos, solo las asistencias.