PROGRAMACIÓN DE SERVICIOS Y PROCESOS

Proyecto Final PGV 1

Sistema de Gestión de Incidencias Cliente-Servidor en Java

DAM-N PGV

Familia Profesional: Informática y Comunicaciones

Profesor: Ruymán O. G.

Fecha: 20/01/2026

Versión 2.2

ÍNDICE DE CONTENIDOS

1. INTRODUCCIÓN.

La empresa PGV Canarias necesita desarrollar un sistema centralizado de gestión de incidencias (SAT) que permita a sus clientes registrar y consultar problemas técnicos de forma remota.

Actualmente, las incidencias se reciben por distintos canales, lo que provoca errores y dificulta su seguimiento. Para solucionar este problema, se implantará un servidor de incidencias capaz de atender a múltiples clientes de forma concurrente y centralizar la información.

2. OBJETIVO.

El objetivo del proyecto es desarrollar una aplicación cliente-servidor en Java, utilizando sockets TCP, que permita gestionar incidencias técnicas de forma concurrente. El sistema deberá atender múltiples conexiones simultáneas, garantizando la coherencia de los datos compartidos.

La correcta implementación de esta funcionalidad básica permitirá alcanzar la calificación mínima de aprobado.

Para obtener una calificación superior, se han establecido distintas ampliaciones opcionales, con puntuación independiente, que permitirán mejorar la nota final hasta alcanzar la calificación máxima.

3. ARQUITECTURA.

Las funciones del servidor son:

- Escuchar conexiones entrantes en un puerto determinado (5000, parametrizado).
- Atender a cada cliente conectado mediante un hilo independiente.
- Gestionar una estructura compartida de incidencias accesible por todos los hilos activos.
- Realizar autenticación básica de usuarios en el servidor.
- Mantener información básica sobre los clientes conectados en cada momento (información de red, nombre de usuario).
- Limitar el número máximo de clientes simultáneos a 10 (parametrizable). Si se supera el límite, el servidor rechazará la conexión mostrando un mensaje y cerrando el socket.
- Liberar correctamente los recursos (sockets, streams, hilos).
- Gestionar la identificación de usuarios y asignación de rol (usuario/administrador) y autorizar comandos según el rol.
- Manejo de los comandos no definidos.
- Validar y normalizar los comandos recibidos.
- La comunicación entre cliente y servidor se realizará mediante mensajes de texto basados en líneas, transmitidos a través de sockets seguros (SSL/TLS).

Funcionalidades para puntuación extra en el servidor:

- Autenticación mediante API externa, usando API key (en cabecera) y envío de credenciales mediante una petición HTTP con JSON (en lugar de autenticación básica en el servidor).
- Uso de un sistema de logging con niveles en lugar de imprimir mensajes sin control por consola.
- Persistencia de datos: almacenar incidencias en disco y recargarlas al iniciar el servidor tras un cierre o reinicio (autoaprendizaje).

El cliente será una aplicación de consola que enviará comandos de texto al servidor y mostrará las respuestas recibidas por el servidor.

- Parametrizar los datos de conexión (IP/host y puerto).
- Establecer conexión con el servidor y gestionar el caso de conexión rechazada (por ejemplo, si se supera el límite de clientes).
- Realizar el inicio de sesión (autenticación básica) enviando las credenciales al servidor.
- Mostrar un menú de opciones en función del rol devuelto por el servidor.
- Enviar al servidor los comandos introducidos por el usuario y mostrar la respuesta.
- Validar mínimamente la entrada del usuario (por ejemplo, evitar enviar comandos vacíos o con parámetros incompletos).
- Gestionar correctamente las excepciones (servidor caído, desconexión, timeout, etc.).
- Cerrar correctamente la conexión y liberar recursos al finalizar

Funcionalidades para puntuación extra en el cliente:

- Uso de un sistema de logging con niveles en lugar de imprimir mensajes sin control por consola.

4. PROTOCOLO

Estos son los comandos que habrá que implementar.

- LOGIN: petición de usuario y contraseña. En función del usuario autenticado, el servidor asignará el rol usuario o administrador.
- El administrador será el único que podrá acceder a los comandos de administración.
- ALTA: crea una nueva incidencia. Se deberá guardar con un id único, texto descriptivo, fecha y hora, estado y usuario que ha realizado el alta.
- LISTAR: muestra el listado de incidencias registradas.
- EDITAR (puntuación extra): permite modificar el texto descriptivo de una incidencia previamente creada.
- CERRAR: cierra una incidencia, cambiando su estado a cerrada.
- CLIENTES (solo administrador): muestra los clientes conectados en ese momento.
- SALIR: finaliza correctamente la conexión con el servidor.

El cliente no debe asumir que tiene permisos: el servidor será quien valide el rol y autorice o deniegue la ejecución de cada comando.

El protocolo de los comandos es el mismo que en tareas anteriores, aunque si se mejora se podrá explicar en la memoria final.

5. EVALUACIÓN.

Parte obligatoria

Aspecto evaluado	Puntos
Ejecución correcta del sistema	1,0
Arquitectura cliente-servidor	0,75
Atención a múltiples clientes (hilos)	0,75
Concurrencia segura de las entidades gestionadas	0,75
Validación de entradas, control de accesos y normalización de comandos	0,75
Gestión de entidades mediante comandos	0,75
Roles (usuario / administrador)	0,25
Memoria técnica	1,0
Defensa del proyecto	1,0

Ampliación

Aspecto evaluado	Puntos
Gestión de excepciones y robustez	0,75
Logging con niveles	0,5
Persistencia de datos	0,75
Autenticación con JSON, API	1,0

La copia total o parcial del proyecto supondrá la no superación del mismo, independientemente de la puntuación obtenida en otros apartados

La superación de este proyecto corresponderá a una nota superior o igual a un 5.