
INFORME PRÁCTICA 2

Esta práctica consiste en la implementación de un sistema distribuido basado en sockets TCP, aplicado a una simulación en la que un robot entrega pedidos dentro de un restaurante. El servidor se encarga de generar y gestionar los pedidos, mientras que el cliente actúa como el robot, que recoge los pedidos y los entrega a las mesas correspondientes.

Para desarrollarse se han empleado las bibliotecas estándar socket, threading, random y time, así como pygame para la representación gráfica. El código está dividido en dos archivos principales: uno para el servidor (servidor.py) y otro para el cliente (pr1_cliente.py).

El servidor implementa funciones de verificación como recoger_pedido, entregar_pedido <número> y estado_pedido, que permiten comprobar el estado de los pedidos y gestionarlos de forma segura. Se validan las entradas recibidas y se responde al cliente con mensajes claros y específicos, incluyendo la notificación de errores en caso necesario.

Se aplicó programación concurrente (Práctica 1) para permitir la generación de pedidos y la escucha de conexiones TCP de forma simultánea, sin bloquear la ejecución del programa.

Uno de los principales desafíos encontrados fue la gestión de las conexiones TCP por parte del cliente. En una primera versión, cada vez que el cliente quería comunicarse con el servidor (por ejemplo, para recoger o entregar un pedido), abría una nueva conexión. Esto causaba una sobrecarga innecesaria de conexiones en el servidor. La solución fue establecer una única conexión persistente desde el cliente al servidor, que se mantiene abierta durante toda la ejecución del programa y se cierra correctamente al final.

Otro problema fue hacer una separación clara entre el cliente y el servidor. Se hizo complicado entender en qué parte tenía que estar ubicada cada función. Este problema se solucionó separando y respetando la responsabilidad de cada componente.

Finalmente, se añadió un registro de actividad (log.txt) donde el servidor guarda todas las interacciones con los clientes, incluyendo conexiones, comandos recibidos, respuestas enviadas y generación de pedidos. Este archivo permite hacer un seguimiento detallado del comportamiento del sistema y facilita el análisis o la depuración en caso de fallos.