

# Relatório Trabalho Prático de Redes de Computadores

Ricardo Mochila – 37762

Inês Veríssimo – 40102

Engenharia Informática

## Descrição do Trabalho

---

O trabalho desenvolvido consiste no desenvolvimento de um servidor multiclient, utilizando a linguagem python.

Foram implementados um sensor, o broker e três clientes, o admin, o publicCliente e o sensor.

O broker é a aplicação principal com capacidade para receber clientes, o seu objetivo é receber dados relativos à qualidade do ar de várias áreas que lhe são enviados pelos sensores.

É também responsável por receber ficheiros do admin com uma atualização de software, verificar a versão enviada e atualizá-la ou não, tendo em conta a versão atual. Guarda as últimas 10 leituras de cada sensor registado.

Public client tem como função simular o comportamento de user. As suas opções passam por escolher um tipo de poluente e receber uma lista dos locais que o contêm, obter a última leitura de um local e tem também um modo publish-subscribe, em que o cliente pode registar um local à sua escolha para receber informações (sobre todos os poluentes) de uma área de preferência.

Admin client é a aplicação que deve ser usada pelos fabricantes de sensores. Comunica com o broker de modo a receber a última leitura de um sensor escolhido, lista os sensores registados no broker, respeitando o formato que era pedido no enunciado, envia um firmware update para o broker e também tem a possibilidade de desativar sensores.

## Desenvolvimento

---

Ao longo do desenvolvimento do trabalho foi necessário tomar decisões que seriam mais tarde importantes para que o mesmo estivesse de acordo com os objetivos.

Foram implementados um servidor, o broker, e três clientes, o admin, o public cliente e o sensor.

### Broker

Através de um socket, o broker permite que se estabeleçam conexões. Assim que um cliente se conecta, o servidor reconhece qual o tipo de cliente que se está a ligar.

Se o cliente for um administrador e enviar um ficheiro de atualização de firmware, o broker deve avaliar o conteúdo do ficheiro e decidir se o envia ao sensor ou se o descarta.

Se o cliente for um novo sensor, o broker regista-o ou, caso este já se encontre registado, adiciona os dados com informação sobre a qualidade do ar enviados pelo sensor num ficheiro json, atualizando-o.

Tem ainda a função de responder aos pedidos do publicClient, percorrendo o ficheiro de dados e selecionando a informação correspondente ao pedido. Quando um cliente se desconecta, o broker apaga-o da lista de clientes conectados através da remoção do socket desse cliente.

A informação recebida dos sensores, é guardada em ficheiros json, por este formato ter como base o uso de dicionários e possibilitar tanto uma maior facilidade em aceder e manipular os dados como uma leitura mais perceptível.

### Sensor

Ao conectar-se a um servidor, os primeiros dados a enviar são específicos a si próprio, como o id e o local onde se encontra o sensor. Os próximos dados gerados são relativos ao componente do ar que esse sensor mede.

Os valores respetivos aos valores da qualidade do ar daquele componente são random mas aproximados do real.

Caso receba um ficheiro de firmware, vai atualizá-lo e os próximos valores chegarão com o firmware atualizados.

## publicClient

No publicClient decidiu-se usar forks para que fosse possível ter 2 processos a correr em simultâneo, nomeadamente, após a subscrição de uma área a ser possível receber notificações e receber o menu para seleccionar outras opções.

Contém um menu com as opções que um user pode executar. Ao procurar por local/tipo de poluente, essa escolha é enviada ao broker e este percorre o ficheiro, devolvendo uma lista com os locais/componentes em resposta à procura.

Caso o publicClient queira subscrever um canal de preferência, esse local é enviado ao broker e em cada sensor desse local, na lista de subscritos é adicionado o socket correspondente ao cliente, para que, a partir desse momento, sejam enviadas notificações através do socket.

## Admin

Esta aplicação também contém um menu. Se a escolha for listar todos os sensores, o admin vai enviar ao broker o pedido e o broker vai devolver-lhe uma listagem. No caso de o administrador apenas querer o último valor recebido de um dado local, o broker responde com a última leitura do sensor.

Para enviar uma atualização de firmware, é pedido um input e a versão será enviada ao broker através de um ficheiro.

Para desativar um sensor, o admin envia ao broker informação de desativação. No servidor é procurado o socket que corresponde ao sensor e é apagado esse socket, eliminando o sensor da lista de sensores registados.

## Conclusão

---

Neste trabalho foi possível consolidar a matéria dada nas aulas, compreendendo melhor como, na prática, funcionam servidores multi client através de sockets e selects.

Deparamo-nos com um problema na execução do nosso programa presente no publicClient, quando se subscreve um local de interesse. Através de um fork criamos 2 processos, mas, ao se receberem notificações do local, o menu deixa de ser visível. Entre as notificações, o utilizador pode inserir um input do menu e este vai ser processado e devolvida uma resposta, mas não é possível mostrar o menu novamente a partir do momento em existe uma subscrição.