UNIVERSIDADE DE ÉVORA

Licenciatura em Engenharia Informática

Estruturas de Dados e Algoritmos II

Docentes: José Saias e Pedro Patinho

2019/2020

Relatório:

Monitorização da Qualidade do Ar

Autores:

Tiago Martinho 35735
João Marques 39996

Índice

1 Introdução

O trabalho consiste na criação de um sistema de aplicações que captam, processam e disponibilizam dados sobre a qualidade do ar numa localização. O sistema é constituído por quatro aplicações denominadas de **broker**, **sensor**, **public client** e **admin client**. Estas aplicações comunicam entre si partilhando vários dados como por exemplo *updates* e dados sobre a percentagem de um certo elemento presente no ar num local de interesse.

2 Escolha de protocolo

Neste foi escolhido o protocolo TCP porque é importante manter a fiabilidade de comunicação entre as aplicações.

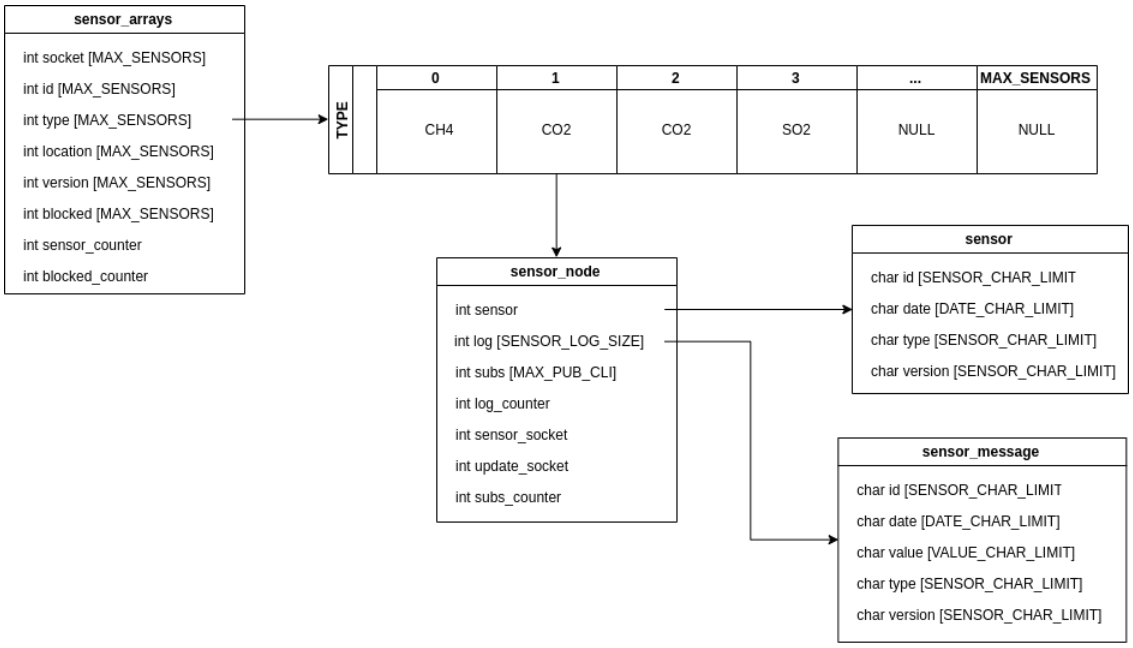
Decidiu-se que a aplicação broker seria o servidor TCP e a outras aplicações seriam clientes TCP.

3 Estruturas Utilizadas

3.1 Sensores

Para guardar a informação de cada sensor utilizou-se estruturas nomeadas ***sensor_node***. Esta estrutura tem as seguintes informações: **ID**, **tipo**, **local**, **versão**, **últimas dez mensagens enviadas** e **clientes subscritos a este sensor**.

Cada um destes nodes é por sua vez guardado numa estrutura ***sensor_arrays*** que guarda os ***sensor_nodes*** por ordem crescente consoante o seu parâmetro (**ID**, **tipo**, **local**, **versão**).



4 Descrição das Aplicações

4.1 Sensor

Para correr a aplicação do sensor tem de ser passar como argumentos o seu **ID**, **tipo**, **local**, **versão**. Usando a seguinte formatação:

```
./sensor ID TIPO LOCAL VERSAO
```

Ao inicializar-se envia a sua informação para o ***broker*** e caso o seu ***ID*** não esteja a ser usado ou já tenha sido disconetado antes, vai então criar uma nova *socket* pela qual irá receber os updates do ***broker***. Usando a função ***select***, se houver um novo *update* cuja a versão seja mais recente, o sensor atualiza.

O sensor de dez em dez segundos (intervalo ajustável) envia os novos dados captados para o ***broker***.

4.2 Broker

Tem como função estabelecer a comunicação entre as aplicações.

O ***broker*** ao receber uma nova conexão é-lhe enviado um carater que irá determinar que tipo de cliente está estabelecer ligação. Caso a conexão seja de um sensor cujo ***ID*** já se encontre em uso ou que tenha sido bloqueado a conexão é imediatamente fechada.

4.3 Public Client

Quando a aplicação se conecta ao ***broker*** envia um carater para se identificar como um *public client*. Seguidamente cria-se uma nova *socket* pela qual o *public client* irá receber os novos dados dos sensores a que subscreveu. Usando a função ***select*** é possível também fazer novos requisitos ao ***broker***, como por exemplo, listar locais onde existem sensores de determinado tipo.

Para listar locais onde existem sensores de determinado tipo:

```
list TIPO
```

Para obter a última leitura de um local:

```
last LOCAL
```

Para subscrever a um local:

```
subscribe LOCAL
```

4.4 Admin Client

A aplicação *Admin Client* envia para o ***broker*** *firmware updates* para sensores de um determinado tipo que o ***broker*** fará chegar aos respetivos sensores.

A aplicação é também a única maneira de desativar sensores.

Para obter a última leitura de um sensor com um determinado*ID*:

```
last ID
```

Para listar sensores registados no ***broker***

```
list
```

Para enviar *firmware updates* para sensores de um tipo:

```
update TIPO VERSAO
```

Para desativar um certo sensor:

```
deactivate ID
```

5 Ficheiros extra

5.1 install.sh

Para compilar corretamente as quatro aplicações sugerimos utilizar este *script* da seguinte forma:

```
sh install.sh
```

5.2 test.sh

Para facilitar o teste deste trabalho incluímos dois *scripts*. Um para utilizadores de ambiente gráfico *gnome* (nomeado *gnome_test.sh*) e outro para utilizadores de ambiente gráfico *xfce* (nomeado *xfce_test.sh*).

```
sh gnome_test.sh
sh xfce_test.sh
```

6 Balanço Final

Alcançou-se o objetivo do trabalho encontrando-se este totalmente funcional. Foi posto em prática o que aprendemos ao longo do semestre o que aprendemos sobre o protocolo TCP. Aprendeu-se a trabalhar com a função *select* e a resolver problemas de maneira criativa.

Ainda assim há espaço para melhorias, como por exemplo, um uso de memória mais eficiente, um algoritmo de pesquisa para o ID mais eficiente (pesquisa binária) e a utilização de *threads* para o *broker* poder satisfazer os requisitos de vários clientes em simultâneo.