



## Redes Comunicações Móveis

---

Projeto - Medição de CO<sub>2</sub> via *Wifi*  
*usando uma rede Ad-hoc*

**Trabalho realizado por:**  
André Mendes, nº78079  
Ruben Condesso, nº 81969  
Miguel Carreiro, nº82012

## Índice

1. Introdução .....	1
2. Especificações do Sistema .....	1
3. Material utilizado.....	2
4. Composição do sistema .....	2
5. Implementação do sistema .....	4
Configurações da rede:.....	4
Configurações dos sockets:.....	5
Configurações dos sensores:.....	5
Redundância do sistema .....	5
6. Montagem do sistema .....	6
7. Limitações do sistema .....	6
Qualidade de recepção da rede <i>Wifi</i> : .....	6
Necessidade de energia elétrica: .....	6
Congestionamento da rede <i>Wifi</i> : .....	6
Congestionamento do <i>RPI 2</i> (que cria a rede <i>ad-hoc 1</i> ):.....	6
8. Resultados obtidos .....	7
9. Conclusão .....	9

## 1. Introdução

Segundo o boletim divulgado pelo WMO (*World Meteorological Organization*), a concentração média global de dióxido de carbono (CO<sub>2</sub>) voltou a crescer e bateu novos recordes em 2016, atingindo o nível mais alto nos últimos 800 mil anos. O aumento nos níveis de concentração do CO<sub>2</sub> nos últimos 70 anos, não tem precedentes na história da humanidade. Os níveis de CO<sub>2</sub> são particularmente elevados nas grandes cidades, pelo que os aglomerados de pessoas que aí vivem, correm diariamente riscos de saúde. Posto isto, faz todo o sentido que seja criado um sistema que meça a qualidade do ar, e que alerte quando o nível de CO<sub>2</sub> passa a ser severamente prejudicial para o homem. Foi no âmbito da disciplina de Redes de Comunicações Móveis, que fizemos a correlação entre este problema atual e a escolha do tema a desenvolver, no projeto da disciplina em causa.

Este projeto é referente, então, ao desenvolvimento de um sistema real, baseado em *Wireless LAN*, que é capaz de detetar se os níveis de CO<sub>2</sub> medidos ultrapassam um limite prejudicial para o ser humano. Caso isto se verifique, o sistema alerta e informa de modo a que se possam tomar as devidas precauções. Esta possibilidade terá de ser desenvolvida no futuro, uma vez que o nosso sistema se restringe apenas à medição do nível de CO<sub>2</sub>, e ao envio da informação pelos vários nós do sistema, desde dos sensores de medição até ao servidor.

As ligações existentes, entre os vários componentes do sistema, são abrangidas por uma rede *ad-hoc*, e a informação recolhida é encaminhada ao longo dos vários nós, utilizando *python sockets* tendo como base uma arquitetura cliente-servidor. Poderão ser adicionados à rede *ad-hoc*, os sensores necessários (ou possíveis) para abranger a área em causa. No nosso caso foi usando apenas um sensor, para efeitos de simplicidade de implementação.

## 2. Especificações do Sistema

A conceção do nosso sistema partir da necessidade de consolidar os vários tópicos que íamos abranger, nomeadamente, os fundamentos das redes *wireless*, que é o tópico principal.

Decidimos utilizar as redes *ad-hoc* como base do nosso sistema. Este modo descentralizado de redes móveis apresenta um elevado nível de flexibilidade e fiabilidade, onde todos os nós funcionam como *routers*. Tivemos em consideração as limitações deste modo, nomeadamente a capacidade limitada de escalabilidade e de área de cobertura.

Para a implementação das redes *ad-hoc*, optamos por usar *raspberrys*. O facto de serem computadores de utilização simples e flexível, de tamanho reduzido e de serem fornecidos pelo Instituto Superior Técnico, verificou-se ser a melhor opção.

Finalmente, no que toca à transferência dos dados decidimos usar *sockets*, por ser um mecanismo com o qual já tivemos contacto no passado e por acharmos que se encaixava perfeitamente nas nossas necessidades. Um *socket* representa um *endpoint* de um fluxo de comunicação, entre componentes numa rede de computadores, sendo que a API fornecida pelas *sockets* permite que os programas as controlem e utilizem, conforme os seus propósitos.

### 3. Material utilizado

Para ir ao encontro das especificações enunciadas no ponto anterior, foi requerido o seguinte material:

- 1 *Raspberry PI 2 Model B*;
- 1 *Raspberry PI 3 Model B*;
- 4 *USB Wifi Adapters*;
- 1 *Breadboard*, juntamente com os fios de ligação e alimentação;
- 1 PC (usando o sistema operativo *Linux*);
- Cabos HDMI e USB, para fazer as ligações necessárias;
- 1 sensor de medição de CO<sub>2</sub>;

### 4. Composição do sistema

De seguida, iremos apresentar a forma como compusemos o nosso sistema. Este encontra-se ilustrado na figura 1:

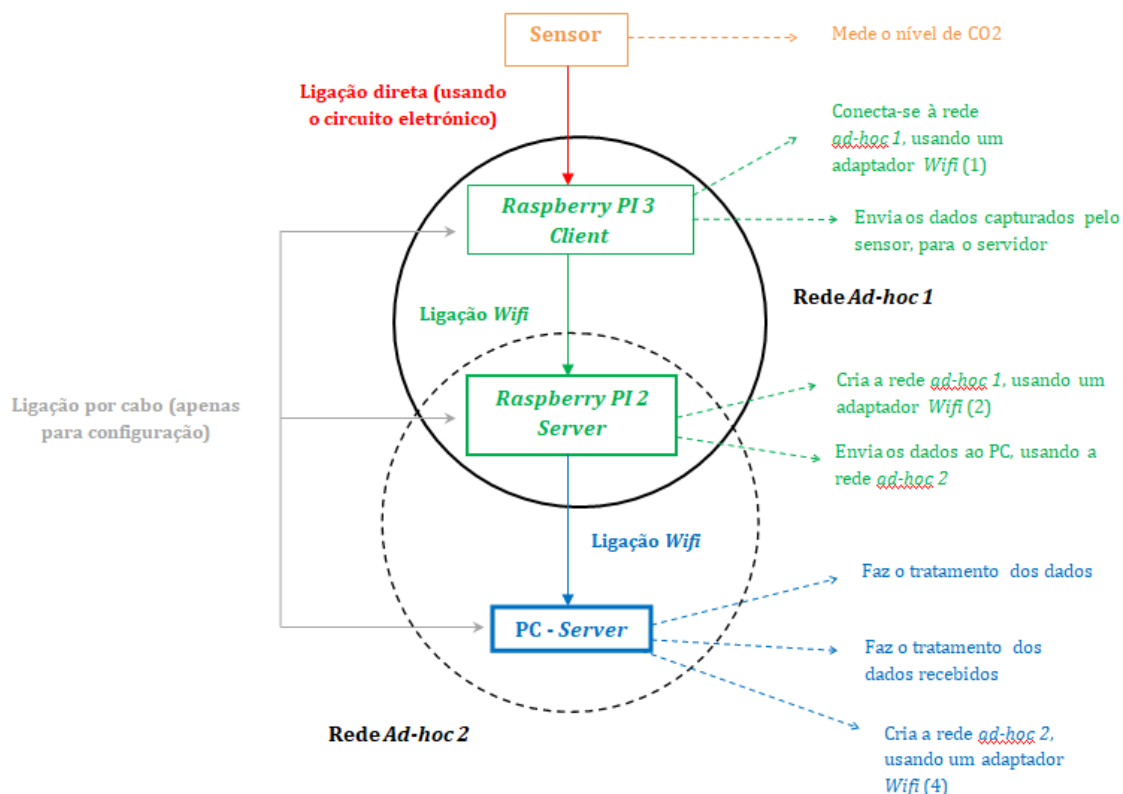


Figura 1 - Composição do sistema implementado, ao nível de rede

Pode-se observar que o epicentro do nosso sistema, passa pelo RPi 2. Este tem a função de criar a rede *ad-hoc* 1, à qual se conectará o RPi 3. Neste ponto de vista, o primeiro age como servidor e este último (RPi 3) agirá como cliente, sendo que o seu objetivo é receber os dados recebidos pelos sensores (no nosso caso só existe 1 sensor), e enviar ao servidor (RPi 2). No fundo, poderiam existir vários clientes ligados ao servidor, onde este seria sempre o único com ligação ao PC. Mas para efeitos de diminuição da complexidade da topologia, foi apenas usado 1 cliente.

Ao receber os dados, o RPi 2 enviará os mesmos ao PC que terá a função de tratar e analisar os dados recebidos. Neste contexto, é usado a rede *ad-hoc* 2, que por sua vez, é criada pelo PC, logo este RPi tem um duplo papel de servidor e cliente.

A figura seguinte é referente à forma como os dados são transferidos ao longo da topologia:

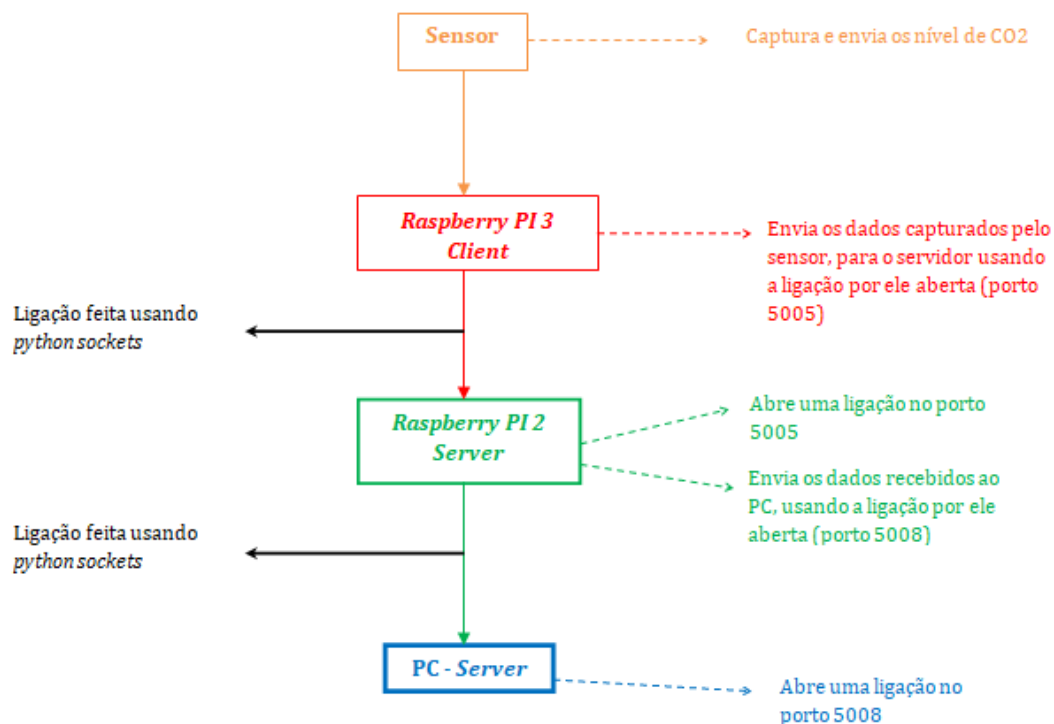


Figura 2 - Sockets usados na transferência de informação, no sistema

Como se pode concluir a partir da figura 2, são criadas 2 ligações para a transferência dos dados capturados, uma entre o RPi 2 e o RPi 3, e outra entre o RPi 2 e o PC. Estas ligações são feitas usando *python sockets*.

## 5. Implementação do sistema

### Configurações da rede:

Com o esboço do sistema feito, a etapa seguinte passou pela configuração da rede *ad-hoc* 1 no RPi2 (*raspberrypi* 2 server). Em primeiro lugar, ligámos o RPi2 ao projetor, de uma sala do Instituto Superior Técnico, para que pudéssemos ativar o SSH, e configurar a *interface* *eth0*. Assim, conseguíamos estabelecer sempre uma ligação por cabo de rede ao nosso PC, e desta forma, a ligação não era interrompida caso houvesse um problema na rede *wifi* (que falaremos a seguir).

#### *Raspberry PI 2:*

Posto isto, passaremos à implementação da rede *ad-hoc* 1, cujas configurações se encontram todas na pasta *Raspeberry2\_Servidor/Configurações\_Ad-hoc1*. Primeiramente, instalámos os programas *dnsmasq* e *hostapd*. De seguida, configurámos as *interfaces* que vão coexistir no RPi 2, nomeadamente a *wlan0* que está responsável pela criação da rede *ad-hoc* 1, e a *wlan1* que será usada para se ligar à rede *ad-hoc* 2. A *eth0* foi também configurada, como referimos anteriormente. Estas configurações dizem respeito ao ficheiro */etc/network/interfaces*, que estão ilustradas no ficheiro *interfaces\_conf*, na pasta referida no início do parágrafo. Posteriormente, configurámos o ficheiro */etc/hostapd/hostapd.conf*, para permitir que o adaptador *wifi* possa criar o AP em causa, como se encontra ilustrado no ficheiro *hostapd\_conf*. Além disso, alterámos o ficheiro */etc/dnsmasq.conf*, para configurar o servidor DHCP, cujas alterações feitas se encontram ilustradas no ficheiro *dnsmasq.conf*. Por fim, tivemos de fazer mais algumas configurações extra, que se encontram explícitas no ficheiro *configs\_extra*. Todas estas alterações foram feitas a partir de uma ligação SSH entre o nosso PC e o RPi 2, usando a ligação direta.

#### *Raspberry PI 3:*

O RPi 3 irá ligar-se à rede *ad-hoc* 1, servindo como cliente, logo tivemos apenas que configurar o ficheiro */etc/network/interfaces*, para indicar à sua *interface* de rede *wlan0* que se conecte ao AP criado pelo RPi 2, referido no parágrafo anterior. Estas alterações estão ilustradas no ficheiro */Raspberry3\_cliente/interfaces\_conf*.

#### *PC-Server:*

Ficava a faltar apenas a criação da rede *ad-hoc* 2, referente ao *PC-Server*. Esta foi criada usando as opções de painel do *Linux*, que se encontram explícitas no ficheiro *PC/create\_AP\_PC.txt*. Tendo todas as configurações da rede feitas, passaremos à explicação dos *sockets* usados na transferência da informação captada pelo sensor.

De notar que, no RPi 2 foi usado o adaptador *wifi* TP-Link TL-WN821N para criar a rede *ad-hoc* 1 e o TP-Link TL-WN321G para se conectar à rede *ad-hoc* 2. No RPi 3 foi usado o adaptador *wifi* TP-Link TL-WN321G.

## Configurações dos sockets:

A configuração das ligações *socket* têm uma implementação bastante simples. O RPi 2 abre uma ligação, usando um certo IP e porto, o RPi 3 conecta-se a essa mesma ligação, e envia-lhe os dados recebidos pelo sensor. O PC abre, igualmente, uma ligação num porto usando um IP, onde o RPi 2 se vai conectar e enviar-lhe os dados. Assim, o PC ficará na posse da informação recolhida, e assim, poderá fazer uma análise dos mesmos.

## Configurações dos sensores:

Houve um obstáculo considerável que nos fez alterar o tipo de sensor a utilizar no projeto, tal situação será esclarecida na [secção 8](#) (Resultados Obtidos). Assim sendo, passámos a usar um sensor que deteta o contacto de metais.

A ligação entre o sensor de metais e o RPi 3, foi feita usando a numeração *BOARD* relativa a este último. O *pin* 1 é usado para fornecer 3.3 V ao sensor, o *pin* 6 corresponde ao *GROUND*, e o *pin* 11 foi usado para a transmissão dos dados digitais do sensor para o *Raspberry*.

Relativamente ao código usado para a transmissão dos dados recolhidos, adicionámos as configurações necessárias ao código do *socket* implementado anteriormente no RPi 3. O GPIO é configurado como INPUT, uma vez que a transmissão dos dados é feita do sensor para o *Raspberry*, e não vice-versa. Assim que o sensor é ativado (ou seja, há uma deteção de metal), o valor de INPUT recebido pelo *Raspberry* é 1. Assim, este sabe quando foi (ou não) detetado metal pelo sensor. Esta informação é depois transmitida até ao PC\_Server.

## Redundância do sistema

Atribuímos robustez e redundância ao sistema, sendo que, caso a rede *ad-hoc* 2 não seja criada (pelo PC), o RPi 2 fica à espera que esta seja criada, e quando assim for, irá tudo operar normalmente. Este foi o último aspeto que tivemos em conta em termos de implementação de sistema, pois é muito comum os sistemas falharem, nem que seja momentaneamente, ou não ligarem pela ordem correta, e assim, é muito importante atribuímos redundância ao sistema. Este ponto, foi também falado nas aulas teóricas da disciplina em causa.

## 6. Montagem do sistema

Em relação à montagem e execução do sistema, iremos explicitar um exemplo de guia de execução:

- 1) Ligar um PC, para servir de *PC-Server* (servidor do sistema);
- 2) No *PC-Server*, criar a AP referente à rede *ad-hoc* 2;
- 3) Executar o código *server\_PC.py*, para criar a *socket-server*;
- 4) Ligar o RPi 2, onde este irá criar automaticamente a rede *ad-hoc* 1, e irá executar o programa da criação da *socket* que lhe diz respeito, de forma automático;
- 5) Ligar o RPi 3, onde este irá conectar-se automaticamente à rede *ad-hoc* 1, e irá executar o programa da criação da *socket* (lado do cliente) que lhe diz respeito, de forma automática;
- 6) Montar o sistema do sensor, na *breadboard*, e ligar ao RPi 3;
- 7) Montagem concluída.

## 7. Limitações do sistema

Este sistema possui algumas limitações, que poderão condicionar a utilização do mesmo num ambiente real.

### Qualidade de receção da rede *Wifi*:

A qualidade do sinal recebido é influenciada por vários fatores, tais como a distância ao *AP*, o número e/ou tipo de obstáculos entre o emissor e o recetor, e as interferências provocadas por outros equipamentos que emitem na mesma frequência. Logo, é um forte condicionante na localização dos sensores.

### Necessidade de energia elétrica:

Para que se possa proceder à recolha de dados, é necessário alimentar cada RPis com energia elétrica. Para tal, existem duas opções: ligar diretamente à tomada elétrica; ou ligar os RPis a uma bateria.

### Congestionamento da rede *Wifi*:

Caso esteja um número elevado de RPis ligados à rede *ad-hoc*, poderá existir um grande volume de dados a serem transmitidos através da rede, congestionando a mesma e podendo provocar atrasos, ou até mesmo perdas no envio dos dados.

### Congestionamento do *RPi 2* (que cria a rede *ad-hoc* 1):

Uma vez que os RPis não possuem grande capacidade de processamento, e que o RPi 2 é responsável não só por receber os dados através do *socket*, mas também por criar e gerir a rede *Wifi*, um elevado número de RPis ligados ao mesmo, poderá originar atrasos no reenvio dos dados para o *PC-Server*. Pode limitar, então, ao número máximo de RPis que podem ser ligados a esta rede.



## 8. Resultados obtidos

Aquando da configuração do sensor em causa, tivemos um contratempo que nos "obrigou" a alterar o sensor a utilizar no trabalho, e por conseguinte, alterou os objetivos definidos anteriormente e o contexto do mesmo. O sensor de CO<sub>2</sub> que nos foi fornecido possui duas formas de transmissão de dados: analógico e digital. No caso do analógico, são enviados os valores exatos da medição efetuada, enquanto no caso do digital, apenas são enviados os bits 0 e 1, onde o bit 0 é enviado quando o valor de CO<sub>2</sub> se encontra abaixo do valor configurado no sensor, e o bit 1 é enviado quando o valor de CO<sub>2</sub> se encontra acima do valor configurado no sensor.

Para o nosso contexto necessitávamos de receber os valores exatos das medições, logo, o único método de transmissão de dados que seria vantajoso era o analógico, no entanto, o RPi 3 apenas lê valores digitais, inviabilizando a utilização do sensor de CO<sub>2</sub>. Para obtermos o valor exato da medição, era necessário usar um ADC (*converter*), que convertesse o valor analógico em digital, e desta forma, o RPi 3 já poderia receber esses valores. No entanto, o Instituto Superior Técnico não tinha disponível este ADC. Posto isto, foi-nos fornecido outro sensor para dar continuidade ao nosso trabalho, sensor este que detetava metais sob contacto.

O objetivo do sistema foi então alterado. Mantivemos toda a arquitetura do sistema, que foi ilustrada anteriormente, porém já não iria existir uma medição de CO<sub>2</sub> por parte do sensor. Este iria detetar o contacto por parte de um metal (ou de um objeto que contivesse metal na sua composição), e enviar para o RPi 3 essa informação. A mesma iria ser transmitida para o RPi 2, que por sua vez, comunicaria ao PC\_Server, onde este último, iria gerar um alarme sonoro. No caso de não se detetar nenhum metal, não se iria transmitir essa informação pelo sistema, de forma a não congestionar as *sockets* utilizadas. Passaremos então à ilustração dos resultados obtidos, para este novo contexto.

Como foi referido anteriormente, as *sockets* são automaticamente criadas com o ligar os RPis, tal como a rede *ad-hoc* 1 (no RPi 2). Apenas no PC\_Server é que é preciso criar manualmente, tanto a respetiva *socket* como a rede *ad-hoc* 2. De notar que o código referente aos dados recebidos pelo sensor, no RPi 3, está no mesmo código que a criação da respetiva *socket*, e logo, também é lançado automaticamente.

Se o sensor detetar um metal, enviará o valor '1' para o RPi 3. Ao receber este valor, o RPi 3 conclui que foi detetado um metal e desta forma, comunicará tal informação ao RPi 2, usando a *socket* criada por este último. Esta parte do sistema encontra-se ilustrada na figura 3.

```

pi@raspberrypi:~/Desktop $ cat socket.log
INFO:root:Socket program running
INFO:root:GPIO - Connecting to pin
INFO:root:GPIO add event callback
INFO:root:Raspberry 3 received sensor data
INFO:root:Raspberry 3 received sensor data
INFO:root:Message sent to Raspberry 2
INFO:root:Raspberry 3 received sensor data
INFO:root:Raspberry 3 received sensor data
INFO:root:Message sent to Raspberry 2
INFO:root:Raspberry 3 received sensor data
INFO:root:Message sent to Raspberry 2
INFO:root:Raspberry 3 received sensor data
INFO:root:Raspberry 3 received sensor data
INFO:root:Message sent to Raspberry 2
INFO:root:Raspberry 3 received sensor data
INFO:root:Raspberry 3 received sensor data
INFO:root:Raspberry 3 received sensor data
INFO:root:Raspberry 3 received sensor data
INFO:root:Message sent to Raspberry 2

```

Figura 3 - Output no RPi 3, quando a detecção de metal por parte do sensor

Na figura 4, é possível verificar o RPi 2 a criar a socket usada para receber os dados do RPi 2, e posteriormente, receber a informação da detecção do metal, por parte do sensor.

```

pi@raspberrypi:~ $ cat socket.log
INFO:root:Socket program running
INFO:root:Client socket created
INFO:root:Binding done, IP: 192.168.0.1, Port: 5005
INFO:root:Listening...
INFO:root:Connection address: ('192.168.0.2', 46340)
INFO:root:Message received: Metal Detected
INFO:root:Message number: 0
INFO:root:Message sent to PC
INFO:root:Listening...
INFO:root:Connection address: ('192.168.0.2', 46342)
INFO:root:Message received: Metal Detected
INFO:root:Message number: 1
INFO:root:Message sent to PC
INFO:root:Listening...
INFO:root:Connection address: ('192.168.0.2', 46344)
INFO:root:Message received: Metal Detected
INFO:root:Message number: 2
INFO:root:Message sent to PC
INFO:root:Listening...
INFO:root:Connection address: ('192.168.0.2', 46346)
INFO:root:Message received: Metal Detected
INFO:root:Message number: 3
INFO:root:Message sent to PC
INFO:root:Listening...
INFO:root:Connection address: ('192.168.0.2', 46348)
INFO:root:Message received: Metal Detected
INFO:root:Message number: 4
INFO:root:Message sent to PC
INFO:root:Listening...

```

Figura 4 - Output no RPi 2, quando o recebimento da informação da detecção de metal

Finalmente, na figura 5, está ilustrado o PC\_Server a receber a informação da detecção do metal por parte do RPi 2, que o levará a gerar um sinal sonoro. A informação é partilhada entre os dois, usando um socket criado pelo PC.

```

andre@andre-X542URR:~/Desktop$ python server_pc.py
('Connection address: ', ('10.42.0.209', 49232))
Metal Detected
('Connection address: ', ('10.42.0.209', 49234))
Metal Detected
('Connection address: ', ('10.42.0.209', 49236))
Metal Detected
('Connection address: ', ('10.42.0.209', 49238))
Metal Detected
('Connection address: ', ('10.42.0.209', 49240))
Metal Detected

```

Figura 5 - Output no PC\_Server, quando o recebimento da informação da detecção de metal, e geração do alarme sonoro

## 9. Conclusão

Em suma, este trabalho permitiu consolidar os conhecimentos de diversos temas abordados nas aulas teóricas, nomeadamente, as *Wireless LAN*, em que consistem, o que permitem fazer, as suas vantagens e limitações. Além desta correlação entre o trabalho e a matéria dada na disciplina, podemos pôr em prática vários conceitos que foram expostas nas 3 principais áreas do nosso curso (LETI): Redes, Informática e Eletrónica. Tudo o que envolveu configurar os RPis e criar as redes *ad-hoc*, diz respeito à área das redes, enquanto, a criação dos *python sockets* é referente à área de Informática, mais concretamente, programação na linguagem *Python*, e finalmente, ligar o sensor ao RPi (montagem do circuito) envolveu conhecimento da área da eletrónica. Posto isto, podemos já concluir que este se tratou de um projeto abrangente e completo.

Ao desenvolver este sistema, muitas vezes tivemos de levar o pensamento de um ambiente local (ao que estamos habituados) para um ambiente real (mundo real), e pensar em como o nosso projeto poderia trazer alguma utilidade ao mundo real, como a professora fez questão de apontar várias vezes ao longo deste semestre. Neste ponto, não podemos deixar de referir a limitação que tivemos ao nível do sensor utilizado. Como já foi explicado no relatório, assim não tivemos matéria suficiente para tornar o nosso sistema em algo mais útil, particularmente no que toca ao nosso objetivo inicial: medir o nível de CO<sub>2</sub> no ar e alertar caso este supere um certo nível prejudicial ao homem. Com a substituição por um sensor de deteção de metal, a utilidade do nosso sistema ficou limitada.

Ainda assim conseguimos estabelecer as várias relações existentes entre os componentes do nosso sistema, tanto a nível de rede como a nível de transferência de dados, nomeadamente, na deteção do metal e no envio dessa deteção ao longo da rede, passando pelas rede *ad-hoc* criadas. Este aspeto foi o ponto mais importante deste trabalho, que foi alcançado com sucesso por nós.

No caso de se dar continuidade a este trabalho, o próximo passo seria mudar o sensor utilizado, e assim voltar à nossa ideia inicial. Permitiria colecionar dados estatísticos referentes às medições capturas pelos sensores utilizados. Esses dados poderiam dar informações importantes sobre o comportamento do CO<sub>2</sub> no ar, e acionar mecanismos que poderia baixar o seu nível, oferecendo assim uma maior utilidade.