



DEPARTAMENTO DE ENGENHARIA INFORMÁTICA

LICENCIATURA EM ENGENHARIA INFORMÁTICA

INTEGRAÇÃO DE SISTEMAS

2019/2020 – 1º semestre

PROJETO – “IPLeiria Smart Campus”

RELATÓRIO FINAL

Gabriel Frazão 2171159@my.ipleiria.pt	Gonçalo Vicente 2172131@my.ipleiria.pt	João Marques 2170937@my.ipleiria.pt	Matheus Martins 2171316@my.ipleiria.pt
--	---	--	---

1. Introdução

Nos dias que correm, a Internet das Coisas (IOT – *Internet of Things*) tem vindo a ser uma tecnologia cada vez mais bem vista pela sociedade. Com o intuito de melhorar as condições de segurança e de bem estar dos utilizadores tal como ajudar na preservação do meio ambiente, a IOT agrupa um conjunto físico de objetos e conecta-os através da Internet. Veículos, eletrodomésticos, sensores, edifícios, entre outras meras coisas do nosso cotidiano interligam-se então numa rede capaz de reunir e transmitir dados de acordo com as preferências de quem a utiliza.

O Campus 2 ou mesmo todo o Politécnico de Leiria poderia ser considerado como um bom cenário de teste desta tecnologia, no que diz respeito à recolha de informações de certos locais do Instituto e ao tratamento desses dados recolhidos.

A solução deverá poder ser flexível o suficiente para poder ser aplicada a qualquer Edifício dos vários Campus do Instituto.

Nesta primeira iteração serão apenas utilizados dados da temperatura e da humidade provenientes da Biblioteca José Saramago. Devido à falta de hardware adequado nesse edifício – sensores –, o grupo terá que lidar com dados provenientes de sensores simulados. Esses dados estarão num ficheiro binário gerado aleatoriamente por um programa em Unix fornecido pelos docentes da Unidade Curricular. Quando o sistema regista os dados no ficheiro também apresenta a seguinte informação na consola:

```
SENSOR ID: 1
TEMPERATURE: 22.42
HUMIDITY: 47.76
BATTERY: 99
TIMESTAMP: 1571568195
```

```
SENSOR ID: 2
TEMPERATURE: 22.83
HUMIDITY: 47.74
BATTERY: 98
TIMESTAMP: 1571568195
```

2. Proposta

Tendo em conta a situação colocada foi proposto então a este grupo de trabalho a elaboração de um projeto baseado na Integração de Sistemas. O objetivo será desenvolver uma plataforma integrada com base na tecnologia IOT a fim de melhorar a eficiência energética, o bem-estar e a vigilância no Campus 2 do Instituto Politécnico de Leiria.

Os dados dos sensores simulados, depois de transformados num determinado formato, deverão ser encaminhados para um sistema no qual vários aplicativos se podem conectar para poderem visualizar, monitorizar e analisar esses dados.

Na implementação deste projeto é pedido que sejam desenvolvidos dois aplicativos de área de trabalho: um terá de ser capaz de mostrar em tempo real os dados da temperatura e da humidade provenientes do sistema num formato de tabela e de gráfico, não havendo necessidade de os armazenar localmente. De notar que, enquanto ativa, a aplicação deverá manter e mostrar os dados extraídos desde o momento em que é aberto até o momento em que é fechado.

O outro aplicativo deverá ser capaz de configurar e acionar alertas de acordo com as condições $<$, $>$, $=$ ou *between* podendo cada um deles ser ativado ou desativado. Toda a configuração necessária deve ser armazenada num arquivo local apropriado e os alertas gerados devem ser compartilhados.

Os dados derivados dos sensores deverão ser persistidos numa Base de Dados.

Para além das aplicações já referidas deverá estar disponível uma API Global que permitirá a utilizadores qualificados, como membros da comunidade académica, aceder e usar, possivelmente em outro tipo de aplicações, tanto os dados como os alertas. Nela terá de ser possível:

- registar um sensor pessoal;
- adicionar novos dados;
- invalidar dados do sensor (no caso de sensores com defeito);
- obter dados do sensor por andar com ou sem intervalo de datas;
- obter uma lista de todos os sensores
- obter dados de alarmes

Opcionalmente também pode incluir ações de armazenamento para os dados dos sensores de temperatura vindos de dispositivos móveis da comunidade – sensores pessoais.

3. Descrição do problema

Uma vez que todo o campus está coberto pela rede Wi-Fi, o grupo vai aproveitar a rede para capturar os dados dos sensores. Tendo em conta que há falta de hardware (sensores), simular-se-á os dados dos sensores através de um ficheiro binário.

Serão então desenvolvidos cinco sistemas independentes. Um dos sistemas, vai simular os dados dos sensores e enviar para um sistema de mensagens em que os outros sistemas estão ligados para que consigam receber a informação simulada dos sensores.

Um outro sistema representará a apresentação dos valores dos sensores em tempo real. Os dados são recebidos através de um Broker e são apresentados em dois gráficos, um referente aos registos da temperatura e outro aos registos da humidade. Os dados são ainda exibidos numa tabela.

Outro dos sistemas a elaborar é uma aplicação de alertas, em que são recebidos dados e emitidos alguns alertas tendo em conta várias condições, por exemplo, quando a humidade ou a temperatura atingem certos e determinados valores.

Será criado ainda um sistema que faz a inserção dos valores enviados pelos sensores na base de dados bem como os alertas enviados pelo sistema de alertas.

Por fim, implementaremos uma Global API, que responde a pedidos HTTP (get, post e delete) feitos pelo utilizador. Na Global API, existe ainda um sistema de autenticação que garante que só utilizadores inseridos na Base de Dados podem fazer pedidos.

4. Resolução dos problemas

Após melhor análise dos problemas apresentados, o grupo optou por começar pela elaboração da arquitetura do sistema.

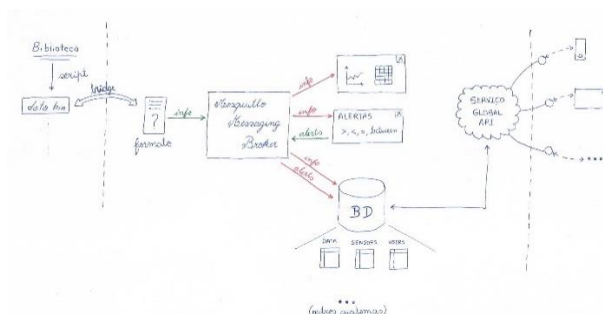


Fig. 1 – Arquitetura do Sistema

Depois de decidir quais iriam ser os tipos dos dados tanto na Base de Dados como no resto do sistema, de se ter decidido que alojar a BD num servidor online seria mais vantajoso e de se ter decidido também que tipo de repartidor de mensagens utilizar, distribui-se o trabalho a ser realizado pelos 4 elementos do grupo.

5. Desenvolvimento dos sistemas e da sua integração

O grupo começa então a implementar as várias aplicações.

5.1. Simulação dos dados provenientes dos sensores e respetiva conversão

Uma vez que não existem sensores físicos foi necessário a utilização de um programa em Unix que simula os dados que viriam dos sensores, para um ficheiro binário.

Este ficheiro é gerado correndo um programa na *bash* de um sistema Linux. Como o Windows tem a possibilidade de criar subsistemas, o grupo criou então um subsistema Linux podendo assim, correr o programa e fazer a conversão do ficheiro.

Após o sistema ler o ficheiro binário hexadecimal, avança para a sua conversão. A conversão foi implementa através de um script em C# que obtém os valores relativos ao registo do sensor. Valores esses que são:

- > id do sensor, um inteiro
- > temperatura, em *float*
- > humidade, em *float*
- > bateria, um inteiro
- > timestamp, representado por um *int long*

Quando a conversão se der por concluída, os valores obtidos são enviados para o repartidor de mensagens num formato XML pelo tópico *info* do repartidor de mensagens.

5.2. Mosquitto Message Broker

O Mosquitto Message Broker foi a tecnologia utilizada para poder ser feita a repartição de mensagens pelos vários sistemas integrados. Este Broker implementa o protocolo MQTT (*Message Queuing Telemetry Transport*) que fornece um método leve de executar mensagens usando um modelo de *Publish/Subscribe*, tornando-o assim adequado para mensagens IOT.

O test.mosquitto.org hospeda um servidor Mosquitto disponível para o público, ou seja, para as aplicações do nosso projeto estarem integradas.

Infelizmente esta tecnologia não funciona na rede da *eduroam* da escola pois os portos que esta tem a usar são bloqueados pela escola

5.3. Aplicação da apresentação dos dados em formato de gráfico e de tabela

O grupo decide implementar uma aplicação em Windows Form no qual são apresentados os dados recebidos pelo Broker enviados pelo sistema de conversão.

A aplicação subscreve-se então pelo *info* ao repartidor de mensagens e trata os dados em forma de gráfico e de tabela, tudo em tempo real.

Foram criados dois gráficos de linhas: um referente aos valores da temperatura e outro que se refere aos valores da humidade. Na tabela é possível visualizar mais informações tais como o id do sensor cujos dados foram registados, o valor da temperatura e da humidade, a bateria do sensor e um timestamp – com a data e hora – daquele registo.

A aplicação é uma Windows Form Application com um componente *TabControl*. Na primeira página foi adicionada o componente *Chart*, onde se podem ver os dois gráficos a cima referidos: o gráfico da temperatura, representado a azul, e o da humidade, representado a laranja, por hora.

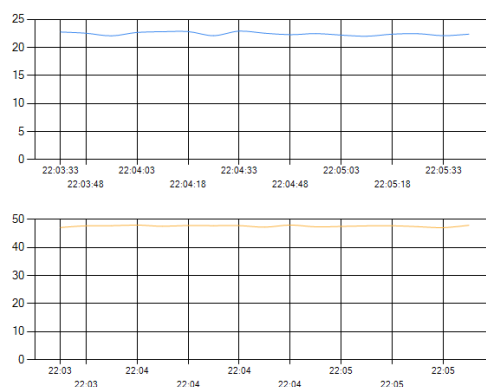


Fig. 2-A – Gráficos atualizados em tempo real para os valores da temperatura e da humidade

Na segunda página implementou-se a tabela com os valores ordenados por ordem descendente de data e hora.

	Id	Temperature	Humidity	Battery	Timestamp
1	1	22.59	47.3	97	19/12/2019 22:05
2	2	22.91	47.44	97	19/12/2019 22:05
1	1	22.1	47.02	98	19/12/2019 22:05
2	2	22.37	47.88	97	19/12/2019 22:05
1	1	22.37	47.67	98	19/12/2019 22:05
2	2	22.45	47.37	98	19/12/2019 22:05
1	1	22.2	47.48	98	19/12/2019 22:05
2	2	22	47.67	98	19/12/2019 22:05
1	1	22.27	47.95	98	19/12/2019 22:04
2	2	22.47	47.32	98	19/12/2019 22:04
1	1	22.91	47.74	98	19/12/2019 22:04
2	2	22.53	47.2	98	19/12/2019 22:04
1	1	22.82	47.81	99	19/12/2019 22:04
2	2	22.1	47.73	98	19/12/2019 22:04
1	1	22.68	47.95	99	19/12/2019 22:04
2	2	22.8	47.51	99	19/12/2019 22:04
1	1	22.53	47.69	99	19/12/2019 22:03
2	2	22.07	47.72	99	19/12/2019 22:03
2	2	22.76	47.07	99	19/12/2019 22:03

Fig. 2-B – Tabela atualizada em tempo real com os valores da temperatura e da humidade

Foram implementadas duas classes: uma referente aos dados dos sensores e outra para a conversão da data do formato *long* para Date Time.

5.4. Aplicação de alertas

O sistema de alertas implementado num Windows Form Application permite a gestão das condições de alerta referentes aos dados que são recebidos, através do Broker, pelo sistema de conversão.

Assim, o sistema de alertas permite criar, atualizar ou remover condições de alerta, sendo que todas essas condições são armazenadas num com um ficheiro XML.

Após a receção dos dados esses são verificados sobre as condições de alerta que o utilizador autenticado criou e que se encontram ativas – uma vez que é possível ter condições de alerta criadas, mas que se encontrem inativas. Caso os registos recebidos estejam de acordo com as condições de alerta, é enviada para o canal *alerts* do Broker a mesma mensagem recebida outrora adicionando apenas uma nova tag “Alerta”, onde irá a mensagem do alerta que foi despoletado.

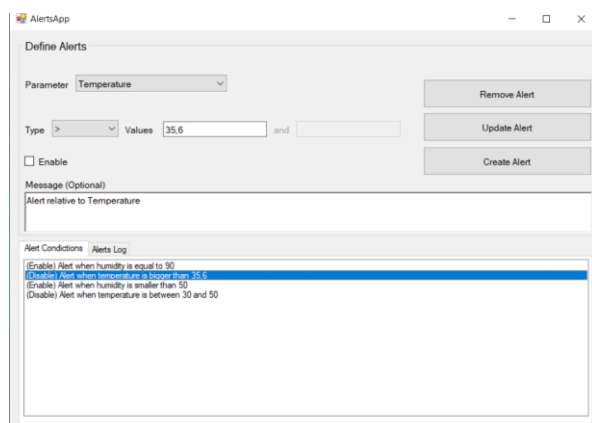


Fig. 3 – Aplicação de alertas

5.5. Base de Dados

Tal como decidido no início da elaboração do projeto, o grupo colocou a Base de Dados num servidor online de seu nome *AppHarbor*.

Nele foram disponibilizadas as opções de conexão para os sistemas que necessitassem da BD a pudessem utilizar.

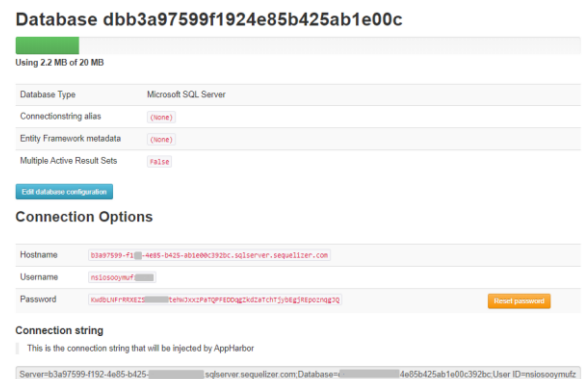


Fig. 4-A – Informações relativas ao servidor online que aloja a Base de Dados

Hostname:

b3a97599-f192-4e85-b425-ab1e00c392bc.sqlserver.sequelizer.com

Username:

nsiosoomufzblta

Password:

KwdbLNFrRRXEZSNW3FuKtehWJxxzPaTQPFEDDqgZkdZaTchTjybEgjREpozngqJQ

Connection String:

Server=b3a97599-f192-4e85-b425-ab1e00c392bc.sqlserver.sequelizer.com;Database=dbb3a97599f1924e85b425ab1e00c392bc;User ID=nsiosoomufzblta;Password=KwdbLNFrRRXEZSNW3FuKtehWJxxzPaTQPFEDDqgZkdZaTchTjybEgjREpozngqJQ;

Depois de ser disponibilizada online, foram criadas 3 tabelas: a *Data*, a *Sensors* e a *Users*.

	Name	Data Type	Allow Nulls
Id	Id	int	<input type="checkbox"/>
Id_Sensor	Id_Sensor	int	<input type="checkbox"/>
Temperature	Temperature	real	<input type="checkbox"/>
Humidity	Humidity	real	<input type="checkbox"/>
Battery	Battery	int	<input type="checkbox"/>
Timestamp	Timestamp	bigint	<input type="checkbox"/>
Alert	Alert	nvarchar(150)	<input checked="" type="checkbox"/>
Valid	Valid	int	<input type="checkbox"/>

Fig. 4-B – Tabela Data

	Name	Data Type	Allow Nulls
Id		int	<input type="checkbox"/>
Name		nvarchar(50)	<input type="checkbox"/>
Localization		nvarchar(100)	<input checked="" type="checkbox"/>
Username		nvarchar(20)	<input checked="" type="checkbox"/>

Fig. 4-C – Tabela Sensors

	Name	Data Type	Allow Nulls
Username		nvarchar(20)	<input type="checkbox"/>
Name		nvarchar(50)	<input type="checkbox"/>
Email		nvarchar(50)	<input type="checkbox"/>
PasswordHash		nvarchar(255)	<input checked="" type="checkbox"/>

Fig. 4-D – Tabela Users

Um sistema para conectar a BD ao repartidor de mensagens foi então elaborado para ser possível receber os valores dos sensores e dos alertas – publicados por outras aplicações do projeto.

Como a BD teria que estar subscrita a dois canais do Broker, *info* e *alerts*, foram implementadas duas funções que recebem ambas como parâmetro a mensagem XML em *string* vinda do Broker.

Na *infoMsg()* os dados provenientes do canal *info* são tratados de acordo com os tipos escolhidos no início do projeto e inseridos na tabela *Data*.

A *alertsMsg()* irá receber os dados que vêm do canal *alerts* – dados esses que já estarão num registo da tabela *Data* à exceção do campo [Alert] preenchido. Será então corrida uma query para atualizar o tal campo [Alert] que se encontrava a *NULL*.

Id	Id_Sensor	Tempera...	Humidity	Battery	Timestamp	Alert	Valid
26	1	22,93	47,24	100	1578656577	NULL	1
27	2	22,15	47,97	99	1578656577	NULL	1
28	1	22,46	47,93	99	1578656582	NULL	1
29	2	22,99	47,42	99	1578656582	NULL	1
30	1	22,47	47,64	99	1578656607	NULL	1
31	2	22,40	47,45	99	1578656607	NULL	1
32	1	22,39	47,25	99	1578656622	NULL	1
33	2	22,48	47,73	99	1578656622	NULL	1
34	1	22,51	47,28	99	1578656637	NULL	1
35	2	22,41	47,14	99	1578656637	NULL	1
36	1	22,62	47,44	99	1578656652	Alert relative to Humidity	1
37	2	22,42	47,74	98	1578656652	Alert relative to Humidity	1
38	1	22,23	47,93	98	1578656697	Alert relative to Humidity	1
39	2	22,39	47,74	98	1578656697	Alert relative to Humidity	1
40	1	22,27	47,34	98	1578656682	Alert relative to Humidity	1
41	2	22,25	47,1	98	1578656682	Alert relative to Humidity	1
42	1	22,07	47,2	98	1578656697	Alert relative to Humidity	1
43	2	22,73	47,06	98	1578656697	Alert relative to Humidity	1

Fig. 4-E – Tabela Data com alguns dados

5.6. REST Global API

A Global API responde a pedidos HTTP. Foi decidido ser implementada em REST devido aos serviços REST serem baseados em rotas. Neste sistema é possível fazer autenticação, o que garante que apenas utilizadores inseridos na base de dados possam fazer pedidos.

Esta está ligada diretamente à Base de Dados SQL Server para melhor gestão direta dos dados.

Em relação aos sensores, é possível obter uma lista com todos os sensores registados, obter um sensor pelo seu id, registar um sensor que pertence à rede de sensores, registar um sensor pessoal e obter a lista dos sensores pessoais do utilizador autenticado.

No que aos alertas diz respeito, é possível obter a lista de todos os alertas, obter a lista de todos os alertas associados a um sensor, obter a lista de todos os sensores com uma determinada localização bem como a lista de todos os sensores dentro de um intervalo temporal. É ainda possível obter uma lista de todos os sensores por localização e dentro de um intervalo de tempo.

Tendo em conta os dados é possível ainda obter todos os dados de todos os sensores, obter a lista de todos os dados de um sensor, obter dados por id, obter a lista de todos os dados de todos os sensores de uma dada localização, obter a lista de todos os dados dentro de um intervalo de tempo, obter a lista de todos os dados dentro de um intervalo de tempo e localização. Ainda é possível registar dados de um sensor pessoal e invalidar dados de um sensor pessoal.

6. Conclusão

Num jeito de conclusão, podemos insinuar que a proposta apresentada pelos docentes da Unidade Curricular foi elaborada com sucesso.

Todos os elementos deste grupo puderam tirar proveito da realização deste sistema e perceber um pouco mais sobre a tecnologia da repartição de mensagens como da leitura de mensagens em formato XML. De considerar também a implementação da REST Global API que fará a ponte para possíveis utilizadores que possivelmente queiram manusear dados deste sistema.

