

Sistemas de Informação Distribuídos

Licenciaturas em Engenharia Informática e Informática e Gestão de Empresas
2019-2020, Segundo Semestre







Deteção de Intrusão e Incêndio em Museus

Mongo DB e Android

Identificação do grupo autor da especificação (Etapa A): **20**

Número	Nome	Foto
82608	Francisco Barros	
20687	Pedro Santiago	
82286	Bin Guan	
82652	Alexandre Ferreira	
79142	Tomás Ferreira	
82453	José Pedro Cruz Fernandes	

Identificação do grupo autor da implementação (Etapas B): 23

Número	Nome	Foto
82493	Miguel Diaz Gonçalves	
83380	Gonçalo Dias do Amaral	
82361	André Freitas	
82946	Pedro Jones	
74278	Dmytro Astashov	
73788	Vitor Manuel Figueira Canhão	

Índice

1	Base de Dados Mongo	7
2	Transporte Broker para Mongo (java)	8
2.1	Periodicidade de Leitura de Sensores e Escrita no Mongo	8
2.2	Utilização de Filtros que impeçam informação de ir para o Mongo	8
3	Estrutura da Base de Dados Mysql	9
3.1	Diagrama relacional.....	9
3.2	Privilégios de Utilizadores.....	12
4	Transporte Mongo para Mysql (java)	13
4.1	Periodicidade de Leitura de Mongo e Escrita no Mysql.....	13
4.2	Utilização de Filtros que impeçam informação de ir para o Mysql	13
4.3	Mecanismo para assegurar que a migração é incremental	15
5	Geração de Alertas	15
5.1	Local/Momento de Detecção.....	15
5.2	Regras para deteção de alertas	16
	Avaliação Global da Qualidade das Especificações	19
6	Implementação	19
6.1	Divergências face ao recebido/especificado	19
6.1.1	Ideias que aproveitaram do outro grupo.....	20
6.1.2	Ideias que abandonaram do próprio grupo	20
6.1.3	Novas ideias introduzidas	20
6.1.4	Código Mongo Implementado (dentro do java)	20
7	Demonstração	21
7.1	Situação 1	21
7.1.1	Dados enviados pelo sensor	21
7.1.2	Alerta registado na tabela de alertas.....	21
7.1.3	Informação Adicional (se relevante)	22
7.2	Situação 2	22
7.2.1	Dados enviados pelo sensor	22
7.2.2	Alerta registado na tabela de alertas.....	22
7.2.3	Informação recebida no telemóvel.....	22

7.2.4	Informação Adicional (se relevante)	22
7.3	Situação 3	23
7.3.1	Dados enviados pelo sensor	23
7.3.2	Alerta registado na tabela de alertas	23
7.3.3	Informação recebida no telemóvel	23
7.3.4	Informação Adicional (se relevante)	23
7.4	Situação 4	23
7.4.1	Dados enviados pelo sensor	23
7.4.2	Alerta registado na tabela de alertas	23
7.4.3	Informação Adicional (se relevante)	24
7.5	Situação 5	24
7.5.1	Dados enviados pelo sensor	24
7.5.2	Alerta registado na tabela de alertas	24
7.5.3	Informação recebida no telemóvel	25
7.5.4	Informação Adicional (se relevante)	25
7.6	Situação 6	25
7.6.1	Dados enviados pelo sensor	25
7.6.2	Alerta registado na tabela de alertas	25
7.6.3	Informação recebida no telemóvel	25
7.6.4	Informação Adicional (se relevante)	26
7.7	Situação 7	26
7.7.1	Dados enviados pelo sensor	26
7.7.2	Alerta registado na tabela de alertas	26
7.7.3	Informação recebida no telemóvel	26
7.7.4	Informação Adicional (se relevante)	26
7.8	Situação 8	26
7.8.1	Dados enviados pelo sensor	26
7.8.2	Alerta registado na tabela de alertas	26
7.8.3	Informação recebida no telemóvel	27
7.8.4	Informação Adicional (se relevante)	27

Deteção de Intrusão e Incêndio em Museus

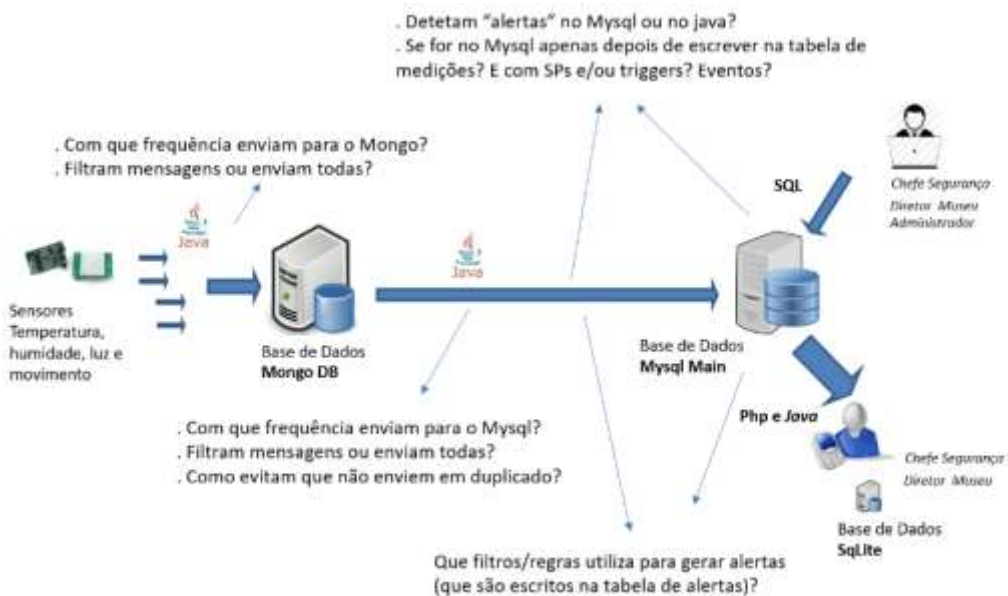
Base de Dados Mysql

Alerta	Utilizador	MediçõesSensores
<<primary key>> ID : int	<<primary key>> EmailUtilizador : varchar(100)	<<primary key>> IDMedição : int
DataHoraMedicao : timestamp	NomeUtilizador : varchar(200)	ValorMedição : decimal(6,2)
TipoSensor : varchar(3)	TipoUtilizador : varchar(3)	TipoSensor : varchar(3)
ValorMedicao : decimal(6,2)		DataHoraMedição : timestamp
Limite : decimal(6,2)		
Descricao : varchar(1000)		
Controlo : boolean		
Extra : varchar(50)		
	Sistema	
	LimiteTemperatura : decimal(6,2)	
	LimiteHumidade : decimal(6,2)	
	LimiteLuminosidade : decimal(6,2)	

Sensor Exemplo Mensagens

```
{ "tmp": "19.30", "hum": "95.00", "dat": "19/4/2020", "tim": "9:50:51", "cell": "228", "mov": "0", "sens": "eth" } (*)
```

Esquema de Importação e Migração



(*) Enquanto o sensor não puder ser reparado, a mensagem é:

```
{ "tmp": "19.30", "hum": "95.00", "dat": "19/4/2020", "tim": "9:50:51", "cell": "228", "mov": "0", "sens": "eth" } Para efeitos de trabalho removam "mov": "0" deve ser substituído por uma vírgula.
```

1 Base de Dados Mongo

Nome bd: MongoSensores

Nome Coleção1(*): DadosSensor

Nome Coleção2(*): _____

Nome Coleção3(*): _____

Exemplo documento coleção1:

Colecao DadosSensor

```
{  
  idMedicao:1  
  temperatura:23.60  
  humidade:95.00  
  luz:388  
  movimento:0  
  timestamp:22/4/2020 14:37:24  
}
```

Exemplo documento coleção3:

Exemplo documento coleção3:

Número de réplicas: 3

(*) O número de coleções é definido pelo grupo, pode optar só por uma ou por um qualquer número.

2 Transporte Broker para Mongo (java)

2.1 Periodicidade de Leitura de Sensores e Escrita no Mongo

<Explicar com que periodicidade o Java recebe informação dos sensores e envia para Mongo>

O programa, em Java, vai recebendo os dados do broker e inserir na BD Mongo o mais rápido possível (depois de eventuais filtrações). Ou seja, as medições devem ser enviadas à medida que chegam do broker.

2.2 Utilização de Filtros que impeçam informação de ir para o Mongo

<Indicar se pretendem que o java implemente filtros que evitem que algum tipo de dados/situação vá para a base de dados MongoDB>

O programa, em Java, deve verificar a formatação dos dados recebidos, bem como possíveis dados em duplicado.

Tendo em conta a formatação de dados, o programa deve verificar caso falte um dado na medição, se for o caso, deve assumir um valor sem informação (Null). Assim, evita-se que ao falhar um sensor (ex: movimento), não há o comprometimento de outros dados importantes (temperatura, humidade, luz).

A filtragem de duplicados (medições com o mesmo timestamp) será feita no transporte do broker para a BD Mongo (java), assim, evita-se que existam dados em duplicado por algum possível erro do sensor.

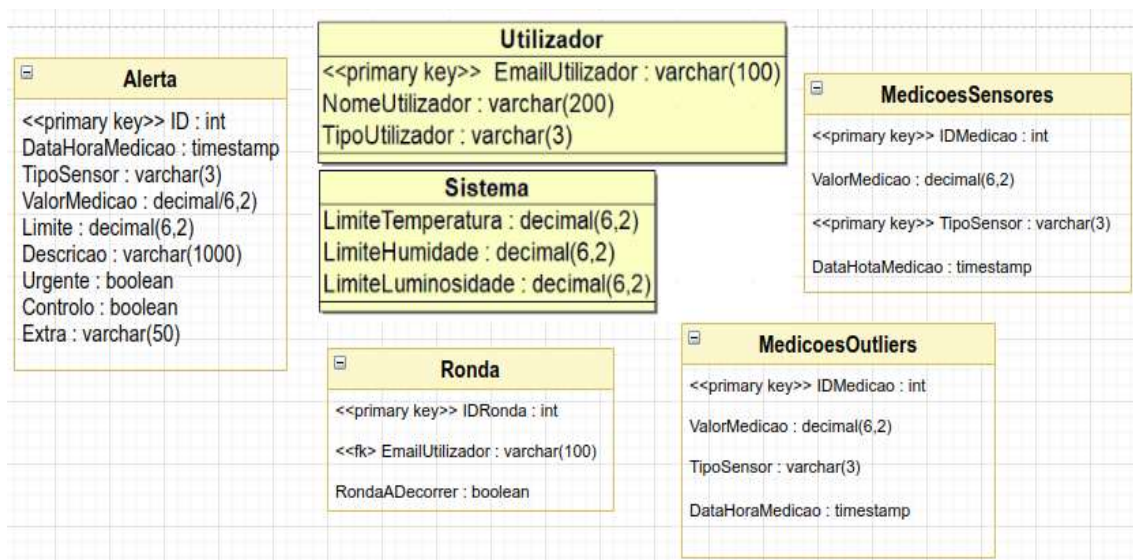
Para fazer a verificação de timestamps, poderá-se utilizar o comando `db.DadosSensor.find(<timestamp>)`. Se for encontrado um dado com o mesmo timestamp do recebido do broker, podemos

assumir que há uma duplicação e o duplicado não deve escrito na BD Mongo.

3 Estrutura da Base de Dados Mysql

3.1 Diagrama relacional

<Diagrama relacional da base de dados Mysql. Podem efectuar alterações à base de dados apresentada no início do documento, no entanto, apenas os 8 campos da tabela Alerta indicados são reconhecidos pela aplicação Android Disponibilizada (os campos Controlo e Extra foram colocados apenas para antecipar eventuais necessidades dos grupos, não são processados pela aplicação Android.)>



Em relação à tabela **MedicoesSensores**, alterámos a primary key para ser o par (IDMedicao, TipoSensor). Fizemos esta alteração para conseguirmos comparar o ID que recebemos do Mongo com os ID's que vão ser inseridos no Mysql. Ou seja, os ID's do Mongo corresponderão aos ID's do Mysql.

Por outras palavras, teremos que para cada informação lida do Mongo serão inseridos 4 registos na tabela MedicoesSensores do Mysql - todas com o mesmo ID mas com TipoSensor diferente.

Exemplo:

Medição no mongo:

```
{
  idMedicao:1
  temperatura:23.60
  humidade:95.00
  luz:388
  movimento:0
  timestamp:22/4/2020 14:37:24
}
```

Medição no Mysql (MedicaoSensores):

IDMedicao <<pk>>	TipoSensor <<pk>>	ValorMedicao	DataHoraMedicao
1	Luz	388	22/4/2020 14:37:24

IDMedicao <<pk>>	TipoSensor <<pk>>	ValorMedicao	DataHoraMedicao
1	Mov	0	22/4/2020 14:37:24
1	Hum	95.00	22/4/2020 14:37:24
1	Tmp	23.60	22/4/2020 14:37:24

Adicionámos o campo Urgente à tabela **Alerta** para sabermos se o evento que gera o alerta é apenas um aviso, ou se é um evento preocupante que deve requerer a especial atenção do Chefe do Museu.

Urgente = false : não muito preocupante (não ultrapassa Limites, mas aproxima-se de tal)

Urgente = true : evento preocupante (ultrapassa os Limites)

Adicionámos a tabela **MedicoesOutliers** para receber os valores discrepantes. Ou seja, que se desviam muito do resto dos valores, e que devem ser eventuais erros.

Também decidimos pôr uma tabela **Ronda**, para não gerar alertas devido às medições de movimento e luz durante as rondas que estejam a decorrer.

RondaADecorrer = true : não deve considerar luz e movimento no momento de gerar alertas.

RondaADecorrer = false : deve considerar todas as medições no momento de gerar alertas.

3.2 Privilégios de Utilizadores

<Quadro com tipos de utilizadores e indicação privilégios sobre tabelas e SPs. Apenas utilizadores relevantes para esta fase >

Tabela	Tipo de Utilizador			
	Administrador	Director de Museu	Chefe de Segurança	
MediçõesSensores	L	L	L	
Utilizador	REL	REL	L	
Alerta	L	L	L	
Sistema	L	L	REL	
Ronda	L	L	REL	
MedicoesOutliers	L	L	L	
Stored Proc.				
AtivarRonda	-	-	X	
ControlarAlerta	X	X	X	

Consideramos que a segurança não é importante para esta fase do trabalho.

Em relação à tabela **Utilizador**, tanto o diretor do museu, como o administrador, poderão fazer alterações nesta tabela. Já o chefe de segurança, apenas poderá consultar a mesma.

No que diz respeito à tabela **Sistema**, consideramos que apenas o chefe de segurança pode alterar estes valores pois será ele que está presente no museu e, como tal, deverá ser o utilizador ideal para ajustar valores de limites de medições (ex: poderá detetar uma maior humidade natural, e, para evitar falsos alarmes, aumenta o valor do limite).

Limitámos a tabela **Alerta** para apenas se poderem fazer leituras sobre a mesma (para qualquer tipo de utilizador). Assim, limitamos ao máximo o potencial erro humano (inserções ou remoções acidentais) - mais do que esta permissão, poderia pôr em causa o sistema de alerta. A mesma lógica se aplica às tabelas **MedicoesSensores** e **MedicoesOutliers**, que apenas deverão poder ser consultadas.

Tendo em conta o SP **AtivarRonda**, terá o propósito de mudar o estado de RondaADecorrer para true (se for começar a ronda) ou false (se estiver a acabar a ronda).

Também criámos um SP **ControlarAlerta** por forma a poder mudar o estado de Controlo para true (se a situação estiver a ser resolvida). Por defeito, Controlo vai estar a false até alguém ativar o SP.

4 Transporte Mongo para Mysql (java)

4.1 Periodicidade de Leitura de Mongo e Escrita no Mysql

<Explicar com que periodicidade o Java recebe transporta informação para o Mysql>

Na nossa opinião, devemos escrever os dados no Mysql logo a seguir a recebê-los do Mongo.

A nossa justificação é que, para além de penalizar o menos possível o tempo de reação do sistema de alarme, também o sistema de migração fica mais impermeável a falhas, porque vai escrevendo a informação no Mysql à medida que a recebe do Mongo. Ex: Caso haja uma falha de energia, temos a certeza que as medições vão sendo escritas no Mysql até ao momento da falha.

4.2 Utilização de Filtros que impeçam informação de ir para o Mysql

<Indicar se pretendem que o java implemente filtros que evitem que algum tipo de dados/situação vá para a base de dados Mysql (tabela Medições)>

Achamos que não é necessário impedir informação de ir para o Mysql, apenas filtrar em que tabela é escrita a informação (duplicação e má formatação de mensagens já são filtrados no transporte Sensor(broker) -> Mongo).

No caso dos outliers e dos alertas, temos que garantir que são escritos na tabela MedicoesOutliers e Alerta, respectivamente. Para filtrar essa informação, consideramos que guardar as últimas 9 medições para poder comparar com a última medição recebida é uma amostra suficiente para uma boa análise.

Caso haja uma medição alarmante, depois de ser verificada que não é outlier, deve ser escrita em MedicoesSensores, bem como Alerta.

Se houver um outlier impede-se essa medição de ser escrita na tabela MedicoesSensores, mas guarda-se esse valor na tabela MedicoesOutliers para futura análise (possível avaria do sensor). Consideram-se outliers as medições que tenham um desvio grande em relação às anteriores medições, ou, que tenham um valor mal formatado (Null).

Exemplo de cálculo para deteção de outlier:

Depois de alguma procura, descobrimos um método estatístico que é eficaz na deteção de um valor outlier num conjunto de dados. (Pode ser consultado aqui: <http://www.mathwords.com/o/outlier.htm>)

Considerando que:

[27] -> medição Temperatura recebida do Mongo

[31,35,40,41,42,43,44,45,48] -> últimas 9 medições (ordem crescente)

Primeiro, temos que calcular:

minimo = 31

primeiro_quartil = 37 i.e. $(31+35+40+41)/4$

mediana = 42

terceiro_quartil = 45 i.e. $(43+44+45+48)/4$

maximo = 48

IQR = $45 - 37 = 8$ i.e. $(1^{\circ}\text{quartil} - 3^{\circ}\text{quartil})$

$$IQR * 1.5 = 8 * 1.5 = 12$$

Agora, podemos definir o intervalo de valores "aceitáveis", ou seja, valores dentro deste intervalo nunca são considerados outliers.

$$\text{intervalo_inferior} = 37 - 12 = 25 \quad \text{i.e.} \quad 1^{\circ}\text{Quartil} - IQR * 1.5$$

$$\text{intervalo_superior} = 45 + 12 = 57 \quad \text{i.e.} \quad 3^{\circ}\text{Quartil} + IQR * 1.5$$

Como o valor da medição recebida [27] está incluído no intervalo [25,57], podemos concluir que não é um outlier.

Este método pode ser usado para Temperatura, Luz e Humidade. Para Movimento não faz sentido fazê-lo pois só assume 2 valores (0 ou 1) e assim, deve-se assumir que medições relativas a Movimento são sempre medições correctas, excepto se for mal formatado (Null).

4.3 Mecanismo para assegurar que a migração é incremental

<Indicar de que forma garantem que não vai informação duplicada para a a base de dados Mysql (tabela Medições)>

Comparamos se para cada ID do Mongo, existem 4 registos correspondentes (com o mesmo ID) no Mysql. Se não existirem os 4 na tabela MedicoesSensores, então, verifica-se também na tabela MedicoesOutliers. Se não houverem 4 no total, podemos concluir que houve problemas com a migração.

5 Geração de Alertas

5.1 Local/Momento de Deteção

Assinalar com X as situações pretendidas (podem ser várias)

No transporte Java X

No Mysql antes de inserir na tabela de medições

De que forma: SP Trigger:

No Mysql depois de inserir na tabela de medições _

De que forma: SP _ Trigger: __

Recorrer a eventos: ____

Vantagens de cada momento de deteção:

Transporte Java:

- **[+]** Mais rápido de todos a poder actuar - será o primeiro a receber as medições da BD Mongo (pode inserir directamente na tabela Alerta, para além de MedicoesSensores)
- **[-]** Teria que ler primeiro a tabela LimiteMedicao para tirar conclusões sobre o alerta.

Mysql antes de inserir:

- **[+]** Mais rápido a detetar do que após a inserção, mais fácil do que no Transporte Java (+ fácil verificar o limite na tabela LimiteMedicao, sendo que já é feito dentro da BD Mysql).
- **[-]** Não é tão rápido a detetar o alerta como no transporte Java - terá que esperar que o java envie a informação para a BD Mysql.

Mysql depois de inserir:

- **[+]** Mais fácil do que no transporte Java a verificar os limites de medição.
- **[-]** Mais lento a detetar do que antes da inserção - as medições têm que ser inseridas primeiro na MedicoesSensores, e só depois poderá ser analisado o possível alerta.

5.2 Regras para deteção de alertas

<Explicar de forma muito clara que regras devem ser implementadas, que tipos de alerta elas despoletam, de que

forma os alertas são guardados na tabela alertas. **Obrigatório dar exemplos**, quer de dados que dão origem aos alertas, quer dos registos que são escritos na tabela alertas. De preferência recorrer a diagramas/ilustrações. Devem indicar explicitamente o nome dos sps que pretendem e/ou o nome das funções java.>

Tipos de alerta:

- **Aviso_Tmp** : Urgente=false, Descrição="Temperatura a aproximar-se do limite"
- **Aviso_Hum** : Urgente=false, Descrição="Humidade a aproximar-se do limite"
- **Aviso_Luz** : Urgente=false, Descrição="Luz a aproximar-se do limite"
- **Alerta_Tmp** : Urgente=true, Descrição="Temperatura passou do limite"
- **Alerta_Hum** : Urgente=true, Descrição="Humidade passou do limite"
- **Alerta_Luz** : Urgente=true, Descrição="Luz passou do limite"
- **Alerta_Mov** : Urgente=true, Descrição="Foi detetado movimento irregular"

Exemplo de registo de um **alerta** (Tabela Alerta):

ID	DataHoraMed	TipoSensor	ValorMed	Limite	Descrição	Urgente	Controlo	Extra
1	22/4/2020 14:37:24	Tmp	51.00	50.00	Temperatura passou do limite	true	false	

Se o valor da medição atingir, ou ultrapassar, o valor do Limite, irá despoletar um alerta. Aplicável para Humidade, Luz e Temperatura.

No caso de Movimento, se a medição assumir o valor 1, será prontamente registado como um alerta - não havendo qualquer aviso antes.

Exemplo de registo de um **aviso** (Tabela Alerta):

ID	DataHoraMed	TipoSensor	ValorMed	Limite	Descrição	Urgente	Controlo	Extra
1	22/4/2020 15:37:24	Tmp	40.00	50.00	Temperaturaproxima-se do limite	false	false	1,2,3,4,5,6,7,8,9,10

Se os valores de temperatura das últimas 5 medições forem todos maiores ou iguais que 80% do valor limite de temperatura, então é emitido um aviso. Se for emitido um aviso, só poderá ser emitido outro após 5 mensagens.

Exemplo:

LT = 50 (Limite Temperatura)

80%LT = 40 (80% do limite de temperatura)

Últimos 10 valor medicao Temperatura = [30,31,35,40,41,42,42,43,43,43]

Segundo o critério acima descrito, neste caso, para estas dez mensagens, irá ser emitido um aviso após a oitava, uma vez que as cinco anteriores são iguais ou maiores que 40. Este critério verifica-se também após a nona e décima mensagem, mas não deve haver logo novo aviso, uma vez que um já foi enviado poucos segundos antes.

Avaliação Global da Qualidade das Especificações recebidas

Avaliação (A,B,C,D,E) : D

Utilize a seguinte escala:

A: - 1 – 5 valores B: 6 – 9 valores C: 10 – 13 Valores D: 14 – 17 valores E: 18 – 20 valores

Análise crítica (clareza, completude, rigor):

Especificação bastante completa e detalhada. Descreve com clareza os principais objectivos da solução.

Salientamos especialmente o método para detecção de outliers proposto, muito bem explicado e com *links* para consulta. Achamos no entanto que não seria necessário um método tão complexo.

Achámos também muito boa a ideia de não repetir um alarme durante um determinado período de tempo. Tal evita o excesso de alarmes seguidos o que pode dificultar a sua leitura.

A detecção antecipada de situações de alarme podia ser melhorada usando um valor de detecção fixo de para cada sensor.

A solução de colocar *null* no valor da medição caso faltem dados nas mensagens recebidas do *broker* não nos parece a melhor. No caso de não se receber valor não se deveria criar uma medição.

Nos privilégios dos utilizadores o Administrador ou o Director de Museu deviam também poder alterar o valor dos limites dos sensores.

A eficiência da consulta dos dados no mongo poderia ser melhorada. Não impede que sejam inseridas leituras com data anterior à última medição registada.

6 Implementação

6.1 Divergências face ao recebido/especificado

<Indicar as divergências relevantes (ignorar pequenos detalhes de implementação) face ao especificado pelo próprio grupo e face ao especificado pelo outro grupo, nomeadamente as que consideram que permitiu chegar a uma solução melhor.>

6.1.1 Ideias que aproveitaram do outro grupo

- Decidimos converter os valores "dat" e "tim" recebidos do sensor para um único valor "dat" com um timestamp. Tal facilita o tratamento de datas no java.

6.1.2 Ideias que abandonaram do próprio grupo

- Abandonamos a ideia de leitura periódica do *broker*, passamos a receber todas as mensagens em tempo real.
- Deixamos de usar a diferença para o valor anterior para detectar possíveis outliers nas medições dos sensores.

6.1.3 Novas ideias introduzidas

- Por sugestão da docente introduzimos um valor mínimo para a temperatura e humidade. Passa a haver também a detecção de valores abaixo de um valor permitido.

- Mudámos a maneira de calcular o valor das medições periódicas. O *mongo2mysql* vai ler todos os documentos com data posterior à da última medição existente no MySQL para um dado sensor. Em seguida aplica uma média móvel exponencial à sequência de valores de modo a eliminar possíveis outliers. No final calcula-se a media aritmética de todas as medições para encontrar o valor a guardar na tabela de medições.

- No caso do sensor de movimento apenas consideramos as medições iguais a 1. Basta uma medição igual a 1 para gerar um alarme.

6.1.4 Código Mongo Implementado (dentro do java)

<Listar todo o código Mongo utilizado dentro do java, quer para importar, quer para exportar. O código tem de ser comentado para que se torne legível para quem sabe uns rudimentos de MongoDB. Não é para colocar código java (excepto se for necessário para perceber o código mongo.>

Faz a ligação ao set de servidores mongo pertencentes à replicaset rs0:

```
mongoClient = new MongoClient(new
MongoClientURI(mongodb://192.168.88.100:27017,192.168.88.100:27018,192.168.88.100:27019/
?replicaSet=rs0));
```

Cria uma ligação à base de dados onde estão as colecções dos sensores:

```
mongoDB = mongoClient.getDatabase("sensores");
```

Cria uma ligação à colecção onde estão os dados do sensor de temperatura:

```
mongoCol = mongoDB.getCollection("temperatura", BasicDBObject.class);
```

Insere um novo documento na colecção dos dados do sensor de temperatura:

```
sensores.runCommand(  
{  
    insert: "temperatura",  
    documents: [ {"tmp":"19.30","dat":"19/4/2020 9:50:51"} ]  
})
```

Para importar usamos um filtro gt (greater than) para filtrar os documentos novos existentes numa colecção. Apenas os documentos com o campo "dat" posterior à data da última medição são considerados.

```
sensores.temperatura.find( { dat: { $gt: "19/4/2020 9:50:51"} } )
```

7 Demonstração

<Ilustrar o funcionamento da aplicação preferencialmente com capturas de imagens. Os exemplos têm de ilustrar todo o tipo de alertas, com a indicação dos respectivos tempos e da forma como visualmente a informação chega ao telemóvel.>

7.1 Situação 1

Tipo de Alerta: Alarme de temperatura excessiva

7.1.1 Dados enviados pelo sensor

```
{"tmp":"28.70","hum":"95.00","dat":"30/5/2020","tim":"20:28:52","cell":"84""mov":"0""mov":"1""sens":"eth"}
```

7.1.2 Alerta registado na tabela de alertas

Sensores.ID	Sensores.senMin	Sensores.senAviso	Sensores.senAlarme
'tmp'	0.00	20.00	25.00

DataHoraMedicao	TipoSensor,	ValorMedicao	Limite	Descricao
'2020-05-30 20:28:52'	'tmp'	'28.70'	'25.00'	'Alarme - Valor acima do permitido'

O tempo entre o envio e o registo na tabela é no máximo 5 segundos.

Informação recebida no telemóvel



7.1.3 Informação Adicional (se relevante)

7.2 Situação 2

Tipo de Alerta: Aviso de temperatura excessiva

7.2.1 Dados enviados pelo sensor

```
{"tmp":"28.50","hum":"95.00","dat":"30/5/2020","tim":"20:35:50","cell":"84""mov":"0""mov":
:"1""sens":"eth"}
```

7.2.2 Alerta registado na tabela de alertas

Sensores.ID	Sensores.senMin	Sensores.senAviso	Sensores.senAlarme
'tmp'	0.00	25.00	30.00

DataHoraMedicao	TipoSensor,	ValorMedicao	Limite	Descricao
'2020-05-30 20:35:50'	'tmp'	'28.50'	'25.00'	'Aviso - Valor perto do limite permitido'

O tempo entre o envio e o registo na tabela é no máximo 5 segundos.

7.2.3 Informação recebida no telemóvel



7.2.4 Informação Adicional (se relevante)

Não é uma situação de alarme, é um aviso que se pode aproximar um alarme.

7.3 Situação 3

Tipo de Alerta: Aviso de temperatura insuficiente

7.3.1 Dados enviados pelo sensor

```
{"tmp":"10.00","hum":"95.00","dat":"30/5/2020","tim":"21:25:10","cell":"84""mov":"0""mov":"1""sens":"eth"}
```

7.3.2 Alerta registado na tabela de alertas

Sensores.ID	Sensores.senMin	Sensores.senAviso	Sensores.senAlarme
'tmp'	20.00	45.00	60.00

DataHoraMedicao	TipoSensor,	ValorMedicao	Limite	Descricao
'2020-05-30 21:25:10'	'tmp'	'10.00'	'20.00'	'Alarme - Valor abaixo do permitido'

O tempo entre o envio e o registo na tabela é no máximo 5 segundos.

7.3.3 Informação recebida no telemóvel



7.3.4 Informação Adicional (se relevante)

Não é uma situação de alarme, é um aviso que se pode aproximar um alarme.

7.4 Situação 4

Tipo de Alerta: Alarme de humidade excessiva

7.4.1 Dados enviados pelo sensor

```
{"tmp":"28.70","hum":"95.00","dat":"30/5/2020","tim":"18:40:14","cell":"84""mov":"0""mov":"1""sens":"eth"}
```

7.4.2 Alerta registado na tabela de alertas

Sensores.ID	Sensores.senMin	Sensores.senAviso	Sensores.senAlarme
'hum'	0.00	75.00	90.00

DataHoraMedicao	TipoSensor,	ValorMedicao	Limite	Descricao
'2020-05-30 18:40:14'	'hum'	'95.00'	'90.00'	'Alarme - Valor acima do permitido'

O tempo entre o envio e o registo na tabela é no máximo 5 segundos.

Informação recebida no telemóvel



7.4.3 Informação Adicional (se relevante)

7.5 Situação 5

Tipo de Alerta: Aviso de humidade excessiva

7.5.1 Dados enviados pelo sensor

```
{"tmp":"28.50","hum":"95.50","dat":"30/5/2020","tim":"18:45:05","cell":"201""mov":"0""mov":"1""sens":"eth"}
```

7.5.2 Alerta registado na tabela de alertas

Sensores.ID	Sensores.senMin	Sensores.senAviso	Sensores.senAlarme
'hum'	0.00	94.00	98.00

DataHoraMedicao	TipoSensor,	ValorMedicao	Limite	Descricao
'2020-05-30 18:45:05'	'hum'	'95.50'	'94.00'	'Aviso - Valor perto do limite permitido'

O tempo entre o envio e o registo na tabela é no máximo 5 segundos.

7.5.3 Informação recebida no telemóvel



7.5.4 Informação Adicional (se relevante)

Não é uma situação de alarme, é um aviso que se pode aproximar um alarme.

7.6 Situação 6

Tipo de Alerta: Aviso de humidade insuficiente

7.6.1 Dados enviados pelo sensor

```
{"tmp":"28.50","hum":"90.00","dat":"30/5/2020","tim":"18:45:05","cell":"201""mov":"0""mov":"1""sens":"eth"}
```

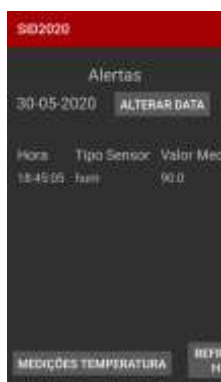
7.6.2 Alerta registado na tabela de alertas

Sensores.ID	Sensores.senMin	Sensores.senAviso	Sensores.senAlarme
'hum'	94.00	97.00	98.00

DataHoraMedicao	TipoSensor,	ValorMedicao	Limite	Descricao
'2020-05-30 18:45:05'	'hum'	'90.00'	'94.00'	'Alarme - Valor abaixo do permitido'

O tempo entre o envio e o registo na tabela é no máximo 5 segundos.

7.6.3 Informação recebida no telemóvel



7.6.4 Informação Adicional (se relevante)

Não é uma situação de alarme, é um aviso que se pode aproximar um alarme.

7.7 Situação 7

Tipo de Alerta: Alarme de luminosidade detectada

7.7.1 Dados enviados pelo sensor

```
{"tmp":"28.50","hum":"95.00","dat":"30/5/2020","tim":"18:57:11","cell":"201""mov":"0""mov":"1""sens":"eth"}
```

7.7.2 Alerta registado na tabela de alertas

DataHoraMedicao	TipoSensor,	ValorMedicao	Limite	Descricao
'2020-05-30 18:57:11'	'cel'	'201.00'	'50.00'	'Alarme - Sensor activado'

O tempo entre o envio e o registo na tabela é no máximo 5 segundos.

7.7.3 Informação recebida no telemóvel



7.7.4 Informação Adicional (se relevante)

7.8 Situação 8

Tipo de Alerta: Alarme de movimento detectado

7.8.1 Dados enviados pelo sensor

```
{"tmp":"28.50","hum":"95.00","dat":"30/5/2020","tim":"19:45:10","cell":"201""mov":"0""mov":"1""sens":"eth"}
```

7.8.2 Alerta registado na tabela de alertas

DataHoraMedicao	TipoSensor,	ValorMedicao	Limite	Descricao
'2020-05-30 19:45:10'	'mov'	'1.00'	'1.00'	'Alarme - Sensor activado'

O tempo entre o envio e o registo na tabela é no máximo 5 segundos.

7.8.3 Informação recebida no telemóvel



7.8.4 Informação Adicional (se relevante)

Não conseguimos testar este tipo de alarme com os dados do sensor uma vez que estes não eram correctos. Inserimos manualmente um documento na colecção do sensor para testar o alarme.