Sistemas de Informação Distribuídos

Licenciaturas em Engenharia Informática e Informática e Gestão de Empresas

2018-2019, Segundo Semestre

Monitorização de Culturas em Laboratório

Mongo DB e Android

Identificação do grupo autor da especificação (Etapa A):

|  |  |  |
| --- | --- | --- |
| Número | Nome | Foto |
| 78109 | Beatriz Cardoso |  |
| 65418 | Diogo Crava |  |
| 68910 | Jorge Lima |  |
| 60850 | Mahomed Hameed |  |
| 65050 | Pedro Coelho |  |
| 78691 | Vasco Branco |  |

Identificação do grupo autor da implementação (Etapas B):

|  |  |  |
| --- | --- | --- |
| Número | Nome | Foto |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

Instruções

Estas instruções são de cumprimento obrigatório. Relatórios que não cumpram as indicações serão penalizados na nota final.

* Podem (e em várias situações será necessário) ser adicionadas novas páginas ao relatório, mas não podem ser removidas páginas. Se uma secção não for relevante, fica em branco, não pode ser removida;
* Todas as secções têm que iniciar-se no topo de página (colocar uma quebra de página antes);
* A paginação tem de ser sequencial e não ter falha;
* O índice tem de estar actualizado;
* Na folha de rosto (anterior) têm de constar toda a informação solicitada, nomeadamente todas as fotografias de todos os elementos dos dois grupos. É obrigatório que caiba tudo numa única página;
* A formatação das “zonas” (umas sombreadas outras não sombreadas) não pode ser alterada;
* O grupo que primeiro edita o documento (Etapa A) apenas escreve até à secção 1.7, e o outro grupo apenas em todas as outras secções.

Índice

[1 Mongo DB 7](#_Toc513111552)

[1.1 Descrição Geral do Procedimento 7](#_Toc513111553)

[1.2 Estrutura da Base de Dados Mongo 8](#_Toc513111554)

[1.3 Periodicidade de Leitura de Sensores e Escrita no Mongo 9](#_Toc513111555)

[1.4 Estrutura da Base de Dados Sybase 10](#_Toc513111556)

[1.5 Periodicidade de Leitura de Mongo e Escrita no Sybase 11](#_Toc513111557)

[1.6 Triggers, SP ou eventos no Sybase (caso relevante) 12](#_Toc513111558)

[1.7 Utilizadores relevantes no Sybase e respectivos privilégios 13](#_Toc513111559)

[1.8 Avaliação Global da Qualidade das Especificações do próprio grupo 14](#_Toc513111560)

[1.9 Implementação 15](#_Toc513111561)

[1.9.1 Código Mongo Implementado (dentro do java) 15](#_Toc513111562)

[1.9.2 Divergências face ao especificado 16](#_Toc513111563)

[1.9.3 Código SQL Implementado 17](#_Toc513111564)

[1.9.4 Divergências face ao especificado 18](#_Toc513111565)

[2 Android e Php 19](#_Toc513111566)

[2.1 Esquema da BD Lite Geral 19](#_Toc513111567)

[2.2 Layout Implementado no Android 20](#_Toc513111568)

Monitorização de Culturas em Laboratório

Um laboratório de investigação de um departamento de biologia necessita de um sistema para monitorizar a evolução de culturas. Mais concretamente, pretende acompanhar a temperatura e luz a que as culturas estão sujeitas, bem como detectar/antecipar potenciais problemas.

Numa estufa estão colocados dois sensores que medem a temperatura e quantidade de luz ambiente (que afecta todas as culturas existentes na estufa).

Periodicamente os investigadores dirigem-se à estufa para efectuarem manualmente várias medições de variáveis (humidade, ph, etc) e registá-las num computador que está localizado na estufa. Cada cultura tem um único investigador responsável e apenas ele pode criar, actualizar e consultar os dados de medições das suas culturas. Esta *protecção de dados* é um aspecto importante do sistema. Nem todas as variáveis necessitam serem lidas e registadas. Para cada cultura o investigador decide quais delas devem ser lidas, e regista no sistema qual o intervalo de valores que considera “normal” para o par variável/cultura.

Por exemplo, para as culturas hidropónicas de pimento e tomate, fazem-se medições do nível de concentração de mercúrio e chumbo. Mas numa cultura de bactérias onde se adicionaram antibióticos o que faz sentido medir é o índice de concentração das bactérias, não faz sentido medir o nível de concentração de mercúrio e chumbo.

**Alertas**

Existem dois tipos de alertas:

a) alertas resultantes das medições das variáveis. O investigador, quando insere manualmente um valor de uma medição, caso o valor ultrapasse os limites será alertado com um aviso (no próprio computador) e com uma mensagem para o telemóvel (por vezes o investigador pede a um colega para efectuar a medição, sendo por isso aconselhável que o alerta não apareça somente no monitor do computador).

b) Alertas resultantes dos sensores de temperatura e luminosidade. O sistema sabe, para toda a estufa, o intervalo de valores de luminosidade e temperatura adequado (igual para todas as culturas). Se o sensor detectar que os valores vão ser ultrapassados deve notificar por telemóvel o investigador.

Cada investigador deverá ter a possibilidade de, através de um telemóvel, monitorizar a evolução da temperatura e luminosidade (não apenas a última leitura, mas a evolução na última hora ou horas) e receber os dois tipos de alertas.

**Registo de Acessos**

É necessário guardar na base de dados (mysql) o registo de todas as operações de escrita sobre todas as tabelas (quais dados foram alterados/inseridos/apagados, quando e por quem) e o registo de operações de consulta apenas sobre a tabela Medições. Esse registo de alterações (*log) é exportado* incrementalmente(apenas informação nova) e periodicamente para uma base de dados autónoma (também mysql). Através dessa base de dados (apenas de consulta) um auditor pode analisar se ocorreram utilizações abusivas dos dados (por exemplo, quem é que alterou limites de temperatura de uma cultura, etc.).

**Diagrama de Use Case Global**



No presente relatório apenas são contemplados os use case “Registo Temperatura”, “Registo Luz”, “Consulta Informação Android”, “Transporte de Dados de Sendor Para Mongo”, “Exportação de Mongo para MySql” e “Exportação de Dados entre Mysql”.

Diagrama de Classes de Suporte à Base de Dados



Sensor

*Exemplo Mensagens*

{"tmp":"22.40","hum":"61.30","dat":"9/4/2019","tim":"14:59:32","cell":"3138","sens":"wifi"}

Esquema de Importação e Migração



# Mongo DB

## Descrição Geral do Procedimento

Nesta segunda fase do trabalho, vai ser especificado todo o processo de migração (escrita dos sensores para MongoDB, migração de MongoDB para mySQL e por fim mySQL para SQLlite) e como será feita a selecção e apresentação dos dados ao investigador, incluindo alertas no telemóvel.

Começando pela aplicação que fará a leitura de dados dos sensores (*PAHO* MQTT) para a base de dados MongoDB, achamos que a mesma deverá transferir todas as leituras (independentemente se corretas ou erradas) para a base de dados MongoDB. Esta estará à escuta constantemente por mensagens do sistema Paho, ou seja, a leitura é constante e feita sempre que houver uma thread disponível de uma Thread Pool com tamanho 5. Essas threads apenas escreveram numa TreeSet que terá mais uma Thread Pool com o mesmo tamanho que estará à espera de dados para enviar para a base de dados MongoDB. No entanto, a TreeSet terá que ter constantemente um tamanho mínimo de 5 leituras para garantir que não existe repetição nas últimas leituras emitidas pelo sensor.

A base de dados MongoDB guardará todas as leituras feitas pelos sensores como mencionado apenas tendo como garantido que não existe repetição das mesmas (verificação feita pela aplicação Java).

A segunda aplicação Java (ligação JDBC entre MongoDB e MySQL), tal como a aplicação Java que faz a ligação dos sensores, não fará qualquer tipo de verificação e apenas a migração periódica dos dados dos sensores presentes no MongoDB.

Finalmente, na base de dados principal MySQL será onde se realizará a maior parte das verificações dos dados dos sensores e lançamento dos alertas por inúmeras razões como: capacidade de os Investigadores poderem customizar os valores que são usados na deteção de erros, centralizar o máximo possível a deteção de erros e lançamento de alertas facilitando assim a customização do sistema sem grande trabalho, possibilidade da customização ser toda feita através da interfaces gráficas que vão ser disponibilizadas ao Administrador e Investigadores, etc.

Para exemplificar este processo, apresentamos o seguinte gráfico (Figura 1).

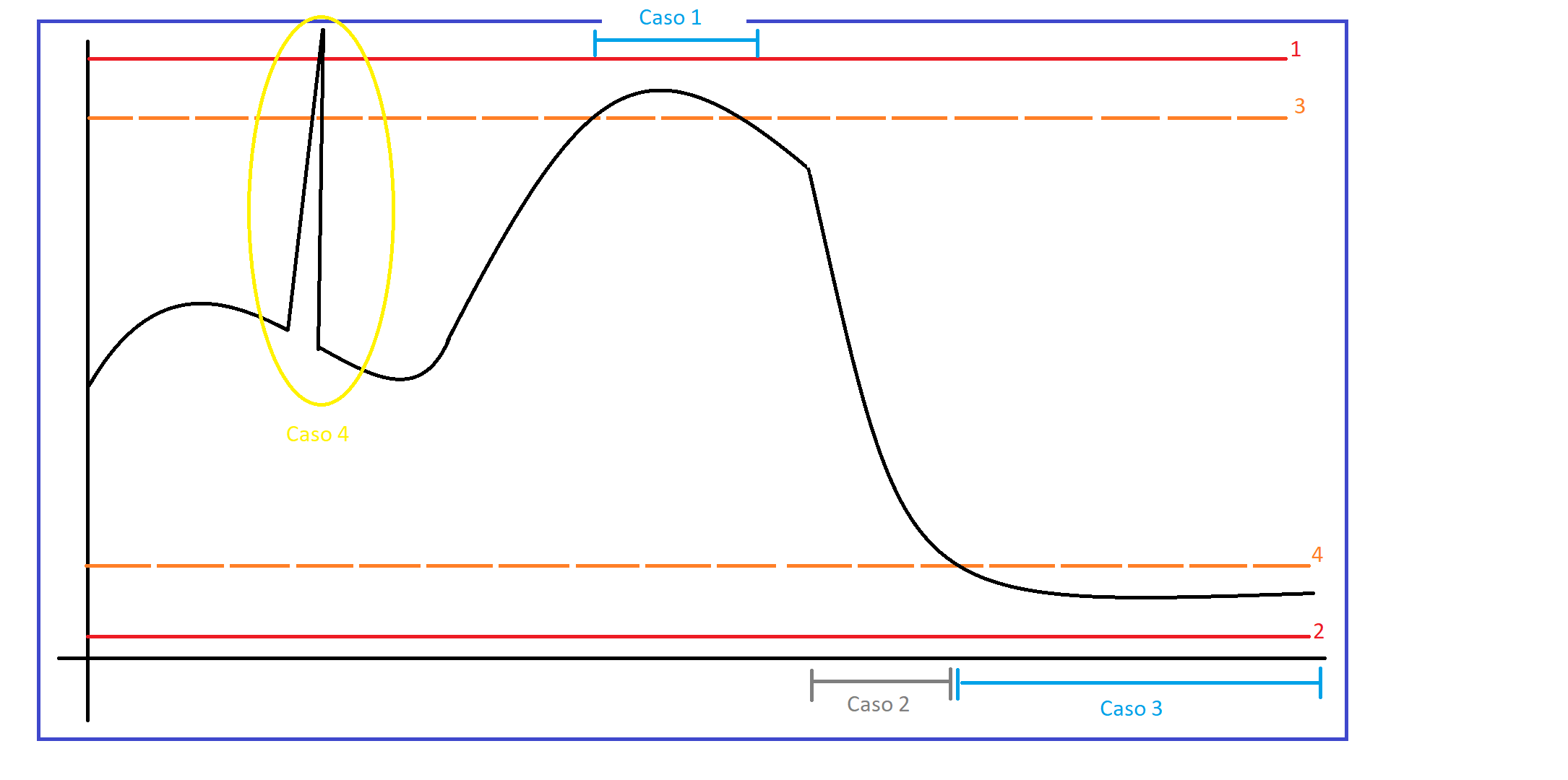


Figura 1 - 1: Limite Superior, 2: Limite Inferior, 3: Limite Superior de prevenção, 4: Limite Inferior de prevenção, Caso 1, Caso 2, Caso 3, Caso 4

Como se pode observar, estão presentes quatro limites, dois deles estabelecidos pelo o enunciado (Limites 1 e 2) **que não podem ser ultrapassados (valores máximos)**, e os outros dois (Limites 3 e 4) foram criados com o propósito de criar “uma barreira de segurança” para o investigador saber caso alguma das variáveis esteja próxima demais de os valores máximos. Todos estes limites vão estar presentes na base de dados principal MySQL, caso os Investigadores necessitam de alterar, fornecendo desta forma flexibilidade ao sistema. (Mais detalhes sobre os alertas na secção 1.3)

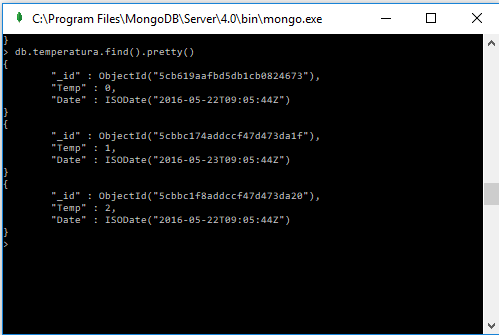
Para além dos limites, temos 4 casos previstos, começando pelo o caso 4, em que o sensor envia uma leitura incorrecta, nós fazemos a verificação da mesma ao delimitar as mudanças súbitas de um valor estabelecido também pelo o investigador, ou seja, caso exista uma mudança drástica de temperatura ou luminosidade acima de o valor definido, este é ignorado pela aplicação. Este valor também vai estar presente na base de dados principal MySQL para caso o valor tenha que ser ajustado. Este valor, para tirar o maior partido do mesmo, não só necessita de conhecimento especializado (ex: como determinar se uma mudança de temperatura é possível ou impossível?) como pode ser dependente do sensor utilizado.

Os restantes casos são referentes a **alertas**. Para o caso 1 e 3, temos um valor que define o intervalo de tempo máximo para que uma variável permaneça num espaço crítico. Após ser ultrapassado o intervalo de tempo estabelecido, o sistema alerta os investigadores. Para o caso 1, a variável apenas permanece no espaço crítico por um tempo inferior ao intervalo máximo, logo o sistema não emite um alerta. No caso 3, este intervalo já é ultrapassado, logo um alerta é emitido.

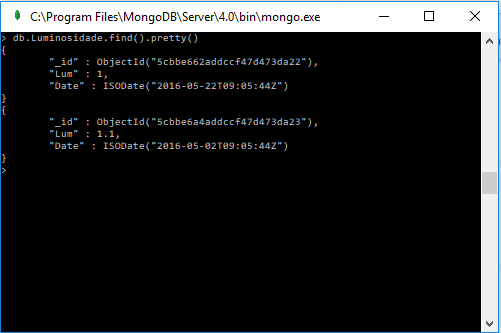
No caso 2, podemos constatar uma descida súbita constante, ou seja, que não ultrapassa o valor de mudança máxima estabelecidos para casos de leituras erradas (caso 4), o que podemos deduzir que não é uma leitura errada do sensor, mas sim uma mudança drástica de temperatura/luminosidade na estufa, o que convém emitir um alerta. (Imaginando casos como portas abertas ou incêndios).

## Estrutura da Base de Dados Mongo

Para a base de dados MongoDB decidimos criar duas coleções, umas para a Temperatura:



E uma para a Luminosidade:



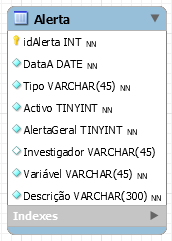
Apesar de a leitura do sensor ser única, ou seja, a temperatura e a luminosidade virem no mesmo objeto JSON achamos que a separação deveria ser feita imediatamente na aplicação Java e escrever em duas coleções diferentes de forma a facilitar a migração de medições entre o MongoDB e MySQL.

## Periodicidade de Leitura de Sensores e Escrita no Mongo

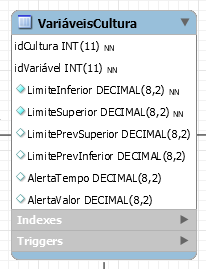
Assumindo que o sensor envia dados com uma periocidade aproximada de 3 segundos, assumimos que haveria um atraso de 15 segundos devido a 5 leituras ficarem constantemente retidas na aplicação Java até a escrita na base de dados MongoDB.

## Estrutura da Base de Dados Mysql

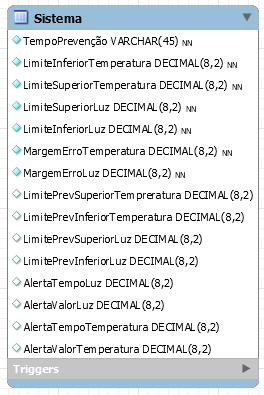
Para além da tabela Alerta que foi adicionada nesta fase também houve tabelas que sofreram alterações.



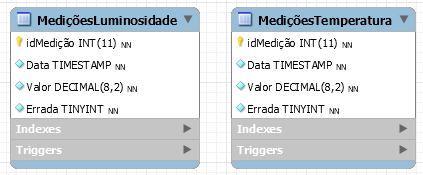
Começando pela nova tabela Alerta, o propósito desta tabela é guardar todos os Alertas lançados aos Investigadores baseados nos casos apresentados na parte 1.1 Figura 1. Os alertas guardaram a data e hora em que são lançados (DataA), o tipo de alerta, ou seja, qual dos casos apresentados na Figura 1, se está activo (caso esteja o(s) investigador(res) em questão continuaram a receber notificações até o mesmo desactivar), se o alerta é geral ou não (caso seja um alerta de temperatura ou luminosidade o alerta é geral e enviado a todos os investigadores), o investigador a que o alerta foi ou vai ser enviado (no caso de ser um alerta geral este campo não é preenchido), a variável responsável pelo alerta e uma descrição mais detalhada gerada pelo sistema.



Continuando para as tabelas alteradas, a tabela VariáveisCultura foi alterada por necessidade de suportar os novos casos analisados na Figura 1. LimitePrevSuperior e LimitePrevInferior são referentes ao novo tipo de limite proposto na parte 1.1, os limites de prevenção. Para além dos novos limites tivemos que acrescentar duas variáveis para suportar o Caso 2 onde procuramos analisar descidas ou subidas “*anormais*” de uma variável, para esse efeito demos o poder aos Investigadores de escolher qual seria o caso mais geral dessas mesmas ocorrências ditando qual seria a mudança máxima (AlertaValor) para um espaço de tempo X (AlertaTempo). Por exemplo, caso a temperatura mudasse 8ºC em apenas 30 segundos e os parâmetros fossem AlertaValor a 5ºC e AlertaTempo 30 segundos então um alerta seria lançado visto que houve uma mudança de 8ºC que por vez é maior do que os 5ºC estipulados.



Tal como a tabela VariáveisCultura, a tabela Sistema teve que adaptar as mesmas variáveis para aplicar à Temperatura e Luminosidade. Para além dessas alterações ainda foram acrescentadas duas outras variáveis, MargemErroTemperatura e MargemErroLuz, que servem para detetar ocorrências do Caso 4 da Figura 1. Elas só ocorrem nesta tabela devido à luz e temperatura serem as únicas subjetiveis a erros visto que são lidas de um sensor. As variáveis MargemErro definem qual é o valor máximo de mudança de uma leitura para outra na tentativa de detetar possíveis leituras erradas dos sensores como mencionada na parte 1.1.



Por fim, foi adicionado a variável Errada (Boolean) às tabelas MediçõesLuminosidade e MediçõesTemperatura para se é uma medição é considerada errada ou não. Nós optamos por não apagar as medições erradas por acharmos que nunca se deve apagar informação e porque o carácter para determinar se estas estão erradas está dependente de valores que podem ser alterados (MargemErroLuz e MargemErroTemperatura). Para além destas razões, saber a frequência com que um sensor lança leituras erradas pode ser uma forma de verificar o bom ou mau funcionamento do sensor.

## Periodicidade de Leitura de Mongo e Escrita no MySql

Como mencionado anteriormente, assumindo que o envio de leituras por parte do sensor é feita aproximadamente de 3 em 3 segundos e que a rapidez e eficácia são características essenciais do sistema para o acesso rápido à informação decidimos meter o Java a fazer uma leitura e exportação com a mesma periocidade para enviar os dados com o menor delay possível.

## Triggers, SP ou eventos no MySql (caso relevante)

|  |  |  |  |
| --- | --- | --- | --- |
| Trigger (Nome) | Operação | Before / After | Descrição |
| MediçõesLuminosidade\_BEFORE\_INSERT | Insert | Before | Antes de introduzir uma medição de luminosidade, é verificado se a mesma está errada ao ver se a diferença da luminosidade anterior da que está a ser inserida é superior à MargemErroLuz presente na tabela Sistema. Caso seja superior, o valor de Errada é mudado para TRUE. |
| MediçõesLuminosidade\_AFTER\_INSERT | Insert | After | **Verificação de Alertas\*** |
| MediçõesTemperatura\_BEFORE\_INSERT | Insert | Before | Antes de introduzir uma medição de temperatura, é verificado se a mesma está errada ao ver se a diferença da temperatura anterior da que está a ser inserida é superior à MargemErroTemperatura presente na tabela Sistema. Caso seja superior, o valor de Errada é mudado para TRUE. |
| MediçõesTemperatura\_AFTER\_INSERT | Insert | After | **Verificação de Alertas\*** |
| Medição\_AFTER\_INSERT | Insert | After | **Verificação de Alertas\*** |
| **Verificação de Alertas** | | | É verificado se existem alertas para ser lançados consoante os Casos 2 e 3 da Figura 1 ou se os Limites Máximos são ultrapassados. Caso haja, é verificado se já existe um alerta igual activo e se não houver é adicionado um novo alerta. Caso não haja, são desactivados todos os alertas do caso a ser verificado. |

## Utilizadores relevantes no Mysql e respectivos privilégios

|  |  |
| --- | --- |
| Tabela | JDBC (Utilizador) |
| MediçõesLuminosidade | Escrita |
| MediçõesTemperatura | Escrita |

No caso do Investigador e Administrador, as permissões permanecem as mesmas.

|  |
| --- |
| Avaliação Global da Qualidade das Especificações recebidas  Avaliação (A,B,C,D,E) : \_\_\_\_\_\_\_\_\_\_\_\_\_  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):** |

## Implementação

### 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.

Tem de ficar claro:

(i)que ideias aproveitaram da própria especificação;

(ii)que ideias aproveitaram da especificação do outro grupo;

(iii)que ideias novas foram introduzidas.

>

### Código Mongo Implementado (dentro do java)

<Listar todo o código Mongo utilizado no processo, quer para importar, quer para exportar. O código tem de ser comentado para que se torne legível para quem sabe uns rudimentos de MongoDB. Fragmentos de código java apenas serão mostrados para dar algum contexto.>

### Código SQL Implementado

<Listar todo o código SQL utilizado no processo de colocação de inserção nas tabelas SQL Anywhere. O código tem de ser comentado para que se torne legível para quem sabe SQL. Os comentários não podem ser redundantes, colocar apenas o essencial. Indicar triggers ou eventos no lado MySQL, se existirem.>

### Tempo Médio

<Indicar o tempo médio que demora um valor do sensor a chegar a base de dados Mysql.>

### Alertas

<Exemplificar alguns alertas gerados automaticamente.>

# Android e Php

## Esquema da BD Lite Geral

<Modelo relacional implementado no Android, tabelas e atributos>

## Layout Implementado no Android

<PrintScreen de um exemplo de interacção>