Sistemas de Informação Distribuídos

Licenciaturas em Engenharia Informática e Informática e Gestão de Empresas

2017-2018, Segundo Semestre

Monitorização de Culturas em Laboratório

Mongo DB e Android

Identificação do grupo autor da especificação (Etapa A):

|  |  |  |
| --- | --- | --- |
| Número | Nome | Foto |
| 62109 | André Vieira | C:\Users\Rodolfo Arnaldo\AppData\Local\Microsoft\Windows\INetCache\Content.Word\andre.jpg |
| 16482 | Paulo Vieira | C:\Users\Rodolfo Arnaldo\AppData\Local\Microsoft\Windows\INetCache\Content.Word\paulo.jpg |
| 69565 | Rodolfo Arnaldo | C:\Users\Rodolfo Arnaldo\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Foto.png |
| 73553 | Rui Tomé | C:\Users\Rodolfo Arnaldo\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Rui.jpg |
| 65345 | Tiago Rodrigues | C:\Users\Rodolfo Arnaldo\AppData\Local\Microsoft\Windows\INetCache\Content.Word\tiago.jpg |

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 na secção 1.1, e o outro grupo apenas em todas as outras secções.

Índice

[1 Mongo DB 6](#_Toc471205788)

[1.1 Descrição Geral do Procedimento 6](#_Toc471205789)

[1.2 Apreciação Crítica sobre a descrição do processo 7](#_Toc471205790)

[1.3 Exemplo de dados armazenadas na colecção 8](#_Toc471205791)

[1.4 Código Mongo 9](#_Toc471205792)

[1.4.1 Implementado 9](#_Toc471205793)

[1.4.2 Divergências face ao especificado 10](#_Toc471205794)

[1.5 Código SQL 11](#_Toc471205795)

[1.5.1 Implementado 11](#_Toc471205796)

[1.5.2 Divergências face ao especificado 12](#_Toc471205797)

[2 Android e Php 13](#_Toc471205798)

[2.1 Esquema da BD Lite Geral 13](#_Toc471205799)

[2.2 Layout Implementado no Android 14](#_Toc471205800)

[2.3 Layout Implementado em php 15](#_Toc471205801)

[2.4 Questões técnicas de implenmentação 16](#_Toc471205802)

Monitorização de Culturas em Laboratório

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

Cada cultura tem um único investigador responsável e apenas ele pode actualizar e consultar os dados de medições das suas culturas. Esta *protecção de dados* é um aspecto importante do sistema.

Sobre cada cultura são regularmente efectuadas (manualmente) medições com base num conjunto de variáveis que variam consoante a cultura. Para cada cultura o sistema conhece o intervalo de valores normal para cada variável, logo, o sistema poderá emitir alertas caso surja um valor anormal.

Por exemplo, para as culturas hidropónicas de pimento e tomate, fazem-se medições do nível de concentração de mercúrio e chumbo. Se, por exemplo, a concentração de chumbo no pimento reduzir significativamente – menos de 25 mg/litro – significa que a planta ajuda a absorver os metais indesejáveis. (*Culturas = pimento e tomate (hidropónico), variáveis = mercúrio, chumbo.)*

Outro exemplo. Numa solução onde convivem bactérias e antibióticos, se o número de bactérias cresce pouco então é porque são sensíveis ao antibiótico (logo, sabemos como as matar se forem prejudiciais). Se o número de colónias de bactérias *Bacillus subtilis,* colocadas junto de antibiótico penicilina, aumentar em mais de 30% em 2 horas é porque o antibiótico não é eficaz. *(Cultura = Bacillus subtilis, variável = penicilina.)*

Existe um sensor que periodicamente *lê* a temperatura e humidade no laboratório. Os dados são registados na base de dados (classe HumidadeTemperatura), e pretende-se que sejam utilizados para emitir alertas (o sistema sabe o intervalo de valores de humidade e temperatura ideal para cada cultura) e para tentar *explicar* eventuais valores anómalos de variáveis (por exemplo, “detecta-se que sempre que a temperatura desce bruscamente – mais do que 5 graus em menos de uma hora – a concentração de ferro no pimento apresenta valores anormalmente baixos”).

Cada investigador deverá ter a possibilidade de, através de um telemóvel, monitorizar a evolução da temperatura e humidade (não apenas a última leitura, mas a evolução da última hora ou horas) e receber alertas relativos a variações bruscas nós valores das variáveis das suas culturas.

É necessário guardar no sybase o registo de todas as operações de escrita sobre todas as tabelas (qua dados foram alterados/inseridos/apagados, quando e por quem) e registo de operações de consulta sobre a tabela Medições. Esse registo de alterações (*log) é exportado* incrementalmente(apenas informação nova) e periodicamente para uma base de dados autónoma (mysql). Através dessa base de dados (apenas de consulta) um auditor pode analisar se ocorreram utilizações abusivas dos dados (por exemplo, verificar se um investigador tentou ler medições de culturas que não as suas, quem é que alterou limites de Temperatura de uma cultura, etc.).

**Diagrama de Use Case Global**



.

Diagrama de Classes de Suporte à Base de Dados Sql Anywhere

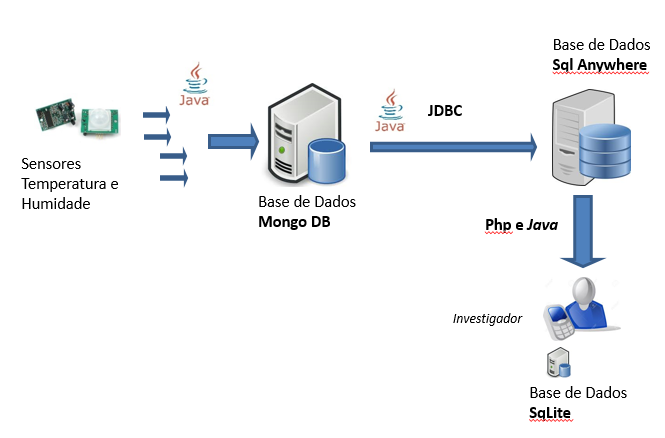


Sensor

*Exemplo Mensagens*

{"sensor":"1"," datapassagem ":"2016/12/12"," horapassagem ":"18:45:24"}

Esquema de Importação e Migração



# Mongo DB

## Descrição Geral do Procedimento

### **Objetivo**

A organização do processo de migração será de forma a que dados enviados pelos sensores sejam armazenados na base de dados MongoDB que posteriormente transitam para a base de dados Sybase. Será a partir ao Sybase que os aparelhos móveis dos investigadores solicitarão as medições aí armazenadas e ainda não visualizadas.

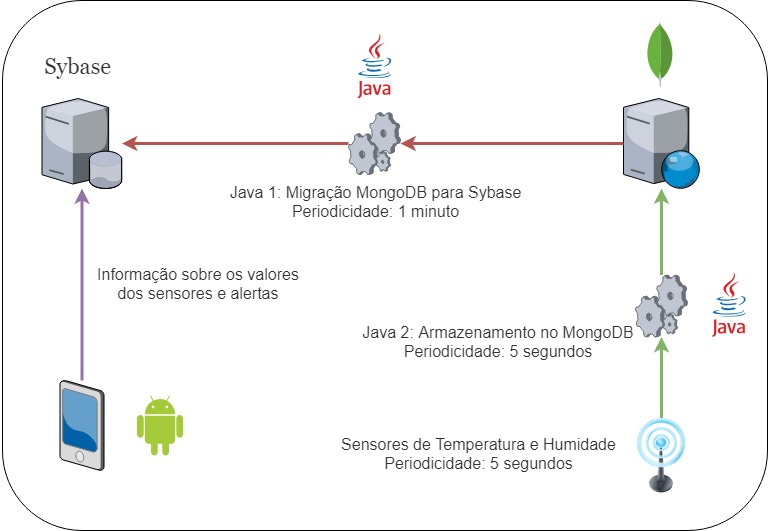


Figura - esquema de processos e agentes desde a sensorização até à leitura

### Java

A componente Java da implementação fará a integração entre os diferentes elementos: Sensores, MongoDB e Sybase, funcionando como intermediário.

Esta componente será estruturada de forma a que sejam executados dois processos distintos, que serão lançados através de dois .jar’s, um para tratar da comunicação entre o MongoDB e o Sybase (J1) e outra que se encarregará de interligar a informação dos sensores com o Mongo DB (J2).

Os dois processos terão as seguintes funções:

* J1 - Este processo será responsável pela migração dos dados do Mongo DB para o Sybase, com uma periodicidade pré-definida e explicada na secção 1.3. Fará também a “tradução” dos dados provenientes do MongoDB e preparará os comandos de inserção no Sybase.
* J2 - Este processo ficará à escuta de dados por parte do sensor e escreverá o respetivo conteúdo no MongoDB, também respeitando uma periodicidade pré-definida.

#### *Configuração dos processos*

Para a configuração dos processos (J1 e J2) é necessário que exista um ficheiro em JSON, para cada processo, com as configurações globais, que será carregado sempre que o programa inicia e traduzido para variáveis do próprio processo. Desta forma, sempre que o processo seja corrido, não será necessário editar código, mas sim apenas o ficheiro de configuração. Para efeitos de simplificação, o ficheiro deverá constar na mesma pasta que o processo (ficheiro jar). Quando à nomenclatura, deverá ser usado os nomes j1.conf, e j2.conf para o processo J1 e J2 respetivamente.

A estrutura a seguir para cada ficheiro de configuração é demonstrada de seguida.

Para o processo J1 (que liga o MongoDB ao Sybase) são necessários os seguintes parâmetros:

* IP do servidor Sybase;
* Porto do servidor Sybase;
* Utilizador Sybase;
* Password do utilizador Sybase;
* IP do servidor Mongo;
* Porto do servidor Mongo;
* Utilizador Mongo;
* Password do utilizador Mongo;
* Periodicidade com que o processo retira os dados do mongoDB e envia para o Sybase.

Para o processo J2 (que liga o sensor ao MongoDB) são necessários os seguintes parâmetros:

* IP do servidor Mongo;
* Porto do servidor Mongo;
* Utilizador Mongo;
* Password do utilizador Mongo.

Opcionalmente, poderão ser adicionados também parâmetros de configuração para o sensor, nomeadamente o tópico e o QOS.

Os ficheiros j1.conf e j2.conf, deverão seguir a seguinte estrutura:

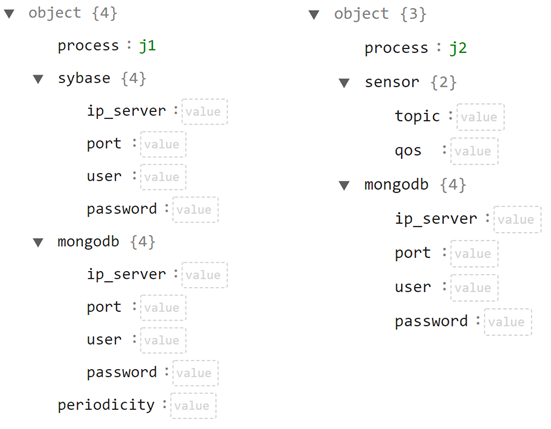


Figura - Estrutura do ficheiro j1.conf (à esquerda) e do ficheiro j2.conf (à direita).

### Processo J1

O processo J1, como referido anteriormente, será responsável pela migração dos dados no MongoDB para o Sybase.

A migração será efetuada respeitando o seguinte procedimento:

1. Selecionar o último membro existente no MongoDB num dado momento;
2. Guardar em memória o id (*timestamp*) desse membro, que será usado como chave de pesquisa;
3. Selecionar todos os registos, presentes no MongoDB, com o id (*timestamp*) igual ou inferior ao do membro selecionado (chave de pesquisa);
4. Preparar uma *string* com os dados para inserção, no formato adequado à execução do comando INSERT para múltiplas linhas – formato:

**INSERT INTO HumidadeTemperatura**

**(dataHoraMedicao, valorMedicaoTemperatura, valorMedicaoHumidade)**

**VALUES (dataHora1, t1, h1),**

**(dataHora2, t2, h2),**

**(dataHora3, t3, h3);**

1. Executar o comando INSERT;
2. Testar o SQLSTATE à saída da execução do INSERT;

* Se SQLSTATE <> 0 (falha durante a operação), fazer ROLLBACK, dar mensagem com SQLSTATE e continuar;
* Se SQLSTATE = 0 (operação executada com sucesso), eliminar todos os membros na base de dados MongoDB com o id (*timestamp*) igual ou inferior ao do membro selecionado (chave de pesquisa);

**NOTA**: É assumido que a eliminação no MongoDB será sempre bem-sucedida, não havendo lugar a erros durante esta operação.

<Nesta secção deverá ser dada uma descrição genérica sobre a forma como decorre o processo de migração:

1. Receber a informação dos sensores e guardá-lo numa base de dados MongoDB;
2. Exportar de **forma incremental** a informação do MongoDB para a bd Sql Anywhere,

A informação apresentada deverá ser suficiente para que o grupo que a receba consiga implementar as várias etapas. Deve ser clara e estar bem estruturada em secções. Cabe ao grupo decidir qual a melhor forma de estruturar a exposição.

Apesar de não ser para escrever código, se o grupo considerar que o grupo que vai implementar pode desconhecer algum aspecto (biblioteca, algoritmo, etc.) pode exemplificar/ilustrar a forma de implementação. Considerar que o grupo que vai implementar tem conhecimentos razoáveis de Java (POO e PCD) e relacional, tem acesso à documentação colocado no E-Learning, e a mais nada.

## Estrutura da Base de Dados Mongo

<Nome da base de Dados e das colecções

Listar algumas linhas exemplificativas da informação guardada na (s) colecção(ões). Usar o comando find().pretty() sem critérios>

## Periodicidade de Leitura de Sensores e Escrita no Mongo

Como definido no projeto, o processo J2 recebe dados provenientes do sensor, de 5 em 5 segundos. Por motivos de fiabilidade (caso o processo J2 vá a baixo), os dados são enviados para o MongoDB sempre que recebidos, não havendo assim qualquer buffer do lado do processo java. Ou seja, sempre que o java recebe um dado do sensor, insere imediatamente no java, e por isso, a periodicidade acaba automaticamente, por ser de 5 em 5 segundos também.

Envio imediato após receção

Envio de 5 em 5 segundos

## Estrutura da Base de Dados Sybase

### Estrutura Geral

### Criação da tabela Alertas

Face à necessidade de criação de alertas para os investigadores relativamente aos valores de temperatura e humidade registados pelos sensores, procedeu-se à alteração da estrutura da base de dados Sybase através da criação da tabela AlertasTemperaturaHumidade.

A tabela terá os seguintes atributos:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nome** | **Tipo** | **Descrição** | **PK** | **Mandatório** |
| idAlerta | integer (auto-inc) | Identificação única do alerta | Não | Sim |
| tipoAlerta | varchar(20) | Tipo de alerta ativado | Não | Sim |
| idCultura | integer | Referência à cultura se necessário | Sim | Não |
| dataHora | dateTime | Data e hora de ativação do alerta | Não | Sim |
| valorReg | decimal(8,2) | Valor de temperatura ou humidade | Não | Não |

### Tipos de Alertas possíveis de detetar

**NoDataAlert** (geral) - é registado quando, num ciclo de envio de leituras do mongoDb, não se verificaram presença de registos dos sensores.

**NoTempAlert** (geral) - é registado quando, em três registos consecutivos, não se verificam valores de temperatura.

**NoHumiAlert** (geral) - é registado quando, em três registos consecutivos, não se verificam valores de humidade.

**HighTempAlert** (relativo a uma cultura) - é registado quando o valor da temperatura de um registo se encontra com uma diferença de 3 graus do limite máximo da cultura.

**HighHumiAlert** (relativo a uma cultura) - é registado quando o valor da humidade de um registo se encontra com uma diferença de 5 unidades do limite máximo da cultura.

**LowTempAlert** (relativo a uma cultura) - é registado quando o valor da temperatura de um registo se encontra com uma diferença de 3 graus do limite mínimo da cultura.

**LowHumiAlert** (relativo a uma cultura) - é registado quando o valor da humidade de um registo se encontra com uma diferença de 5 unidades do limite mínimo da cultura.

**IncTempAlert** (geral) - é registado quando, nos últimos 60 registos consecutivos, se verifica uma subida de temperatura de 1 grau.

**DecTempAlert** (geral) - é registado quando, nos últimos 60 registos consecutivos, se verifica uma descida de temperatura de 1 grau.

**IncHumiAlert** (geral) - é registado quando, nos últimos 60 registos consecutivos, se verifica uma subida de humidade de 2 valores.

**DecHumiAlert** (geral) - é registado quando, nos últimos 60 registos consecutivos, se verifica uma descida de humidade de 2 valores.

## Periodicidade de Leitura de Mongo e Escrita no Sybase

O processo J1, é responsável, como indicado anteriormente, pelo processo de retirar os dados do MongoDB e inserir no Sybase. Para este caso decidimos que o processo J1, deverá retirar os dados do MongoDB de minuto em minuto, exportando assim para o sybase (no melhor caso possível), 12 registos de cada vez. Desta forma, garantimos que os dados estão sempre bastante atualizados, mas com a garantia também que não são efetuadas demasiadas querys ao sybase de forma a gerar um aumento significativo da memória disponível.

Puxa os dados de 60 em 60 segundos

Envio imediato após receção

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

<Especificar que trigger ou SO pretendem que sejam implementados >

## Utilizadores relevantes no Sybase e respetivos privilégios

Como referido no relatório anterior, existirá um utilizador que terá a função de escrever dados na tabela “HumidadeTemperatura”, sendo este designado como MongoDB, que será utilizado pelo processo Java 1 (ligação entre MongoDB e Sybase), para transporte de dados. O mesmo, como especificado no relatório anterior, irá precisar de apenas a permissão de INSERT na tabela” HumidadeTemperatura”.

|  |  |
| --- | --- |
| **Permissões** | **MongoDB** |
| **Tabelas** |  |
| Investigador | - |
| Cultura | - |
| Variaveis | - |
| VariaveisMedidas | - |
| Medicoes | - |
| HumidadeTemperatura | INSERT |
| Tabelas Logs | - |

## Avaliação Global da Qualidade das Especificações do próprio grupo

|  |
| --- |
| 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

### 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 >

### Divergências face ao especificado

<Indicar as divergências relevantes (ignorar pequenos detalhes de implementação) face à especificação recebida, nomeadamente as que consideram que permitiu chegar a uma solução melhor.>

### 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 Sql Anywhere, se existirem>

### Divergências face ao especificado

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

# 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>