Nos procedimento de alteração dos parametros adicionais temos de fazer uns checks a alguns valores que têm de ser maior que 0.

Na discrição do factor para calculo de periodicidade dizer 'enquanto maior este numero, mais rapido a periodicidade se aproxima do tempo maximo permitido em função de SegundosAberturaPortasExterior.

Devemos parametrizar quantos sensores de temperatura no nosso labirinto, e pk?

Pk para fazer uma analise de outliers temos de ter um array de elementos para avaliar. Isto significa que no mazeManage,

na entrada lastTemp, vamos ter tbm para cada um dos sensores o ultimo array de temps gerado (pk o java pode ir abaixo e voltar).

Mas para termos os numeros de sensores temos de controlar, dentro dos possiveis, quantos permitimos e que nomencultura vão ter

(idealmente o nome destes é um INT que pode estar traduzido para o sensor dentro da gama que aceitamos). Portanto temos de parametrizar

logo em SQL quantos permitimos. Isto também permite expandir o número de sensores num labirinto mais tarde.

Quando é feito um fetch de uma coleção de dados de temperatura, o primeiro passo é correr sobre todos e determinar o que pertence a cada sensor

e os que não pertecem a um sensor válido. Devem ser feitas as respetivas validações com o mazeManage (ir ver se a entrada esta criada, ver numExp, etc). Em caso de criarmos

entrada no mazeManage, determinamos logo o tamanho do vetor array a ser usado para despistar outliers (explicação no final).

Depois, para cada um dos grupos de dados com respeito aos sensores, corre uma thread que faz o tratamento dos dados la dentro.

Esta thread deve tbm, à medida que vai passando pelos dados, fixar logo o ultimo movimento 'lastMov'.

Pode ser feito um countdownlatch quando se lança as threads para a threads principal saber quando acordar.

Depois de lançar as outras threads é gerado os statements sql para os dados de erro, posto numa lista, e a threads fica em sleep no countdownlatch.

O tratamento dos dados consiste em ir buscar o array ao mazeManage{lasTemp{sensor x: Array}}, e ir vendo por ordem dos dados aqueles que são dados válidos e que são ou não

outliers (com refeed incondicional do array de temps (isto tbm pode ser parametrizado)). Para os elementos considerados inválidos e outliers é sempre gerado um 'erroexperiencia'

para enviar. Para os elementos que estão bons é verificada primeiro se estao dentro da gama de limite de temperatura estabelecido pelo AppAdmin em SQL e depois criada uma 'medicaotemperatura'.

No final, cada uma destas Threads retorna (lista-comandos-sql, map<numero sensor, array de temps>).

As listas são appended à lista inicial com os erros, criada pela threads principal.

Estando as sub threads terminadas, a thread principal entra num estado de operações transacional.

Ela executa todos os sql statements da lista, atualiza no mazeManage o lastMov com o objeto da ultima temp enviada, e os arrays para cada um dos sensores.

Aqui termina as operações transacionais.

Sleep periodicidade

Repeat

Como é feito o calculo dos outliers?

Temos de admitir que no preenchimento inicial dos arrays aceitamos todas as temperaturas que chegarem.

No pior dos casos a experiencia fica invalida pk o investigador sentiu que devia terminar a mesma por ter temperaturas altas.

Esta interrupção pode servir para a equipa tecnica e os investigadores colaborarem para o despiste de sensores defeituosos, sem que isso

comprometa a vida dos ratos.

Posto isto, o tamanho do array deve ser calculado em função do numero de segundosdeaberturaportasexterior. Temos de fazer testes para ver qual a velocidade de dados

que recebemos para um sensor individual.

Suponhemos que para o sensor 1 recebemos 2 dados por segundo. Se o array tivesse 20 posições demorariamos 10 segundos a preencher o array.

Mas 10 segundos pode ser mais que aberturaportasexterior.

Suponhemos o seguinte, DPS = dados por segundo, T = tamanho do array, TPA = tempo para preencher array, SAPE = SegundosAberturaPortaExterior

A nossa preocupação é que TPA<SAPE, e sabemos que TPA = T/DPS, isto é T/DPS < SAPE. Ou seja, T < SAPE \* DPS.

O tamanho do array [T] tem de ser menor que o produto entre segundosaberturaportaexterior e dadosporsegundo (para um sensor).

Só temos de fazer um teste para ver quantos dados por segundo um sensor de temperatura manda e o resto é calculado de acordo com as variaveis locais.