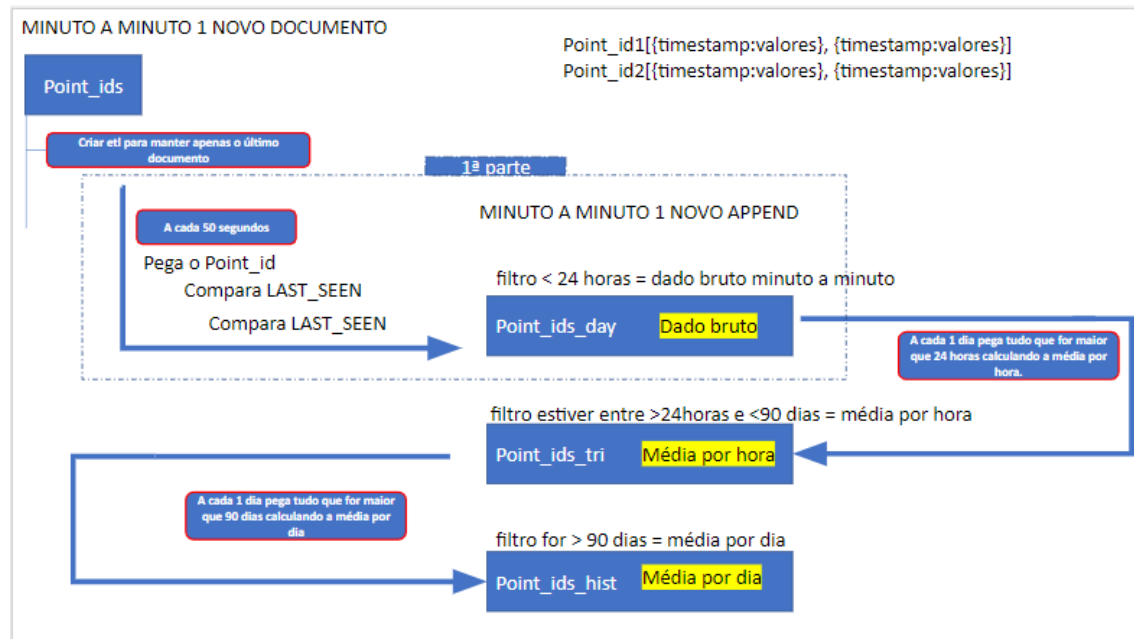


## Especificação Funcional: ETL's Dados históricos – Sensores

Realizada por Denise Proença e Sara Santana

Prazo para finalização: Cada tarefa apresenta sua data limite. A soma de todos os prazos resulta no dia **28/04/2020 às 18h** se uma outra pessoa que entende de programação ajudar com os agendamentos dos processos.

### Arquitetura:



Serão 4 ETL'S:

## 1º Etapa – Point\_ids\_day – Prazo limite é 17/04/2020 às 18 horas



- Extração dos dados da collection point\_ids:

```
point_ids
```

Onde os dados são entregues dessa forma:

```
{
  "name_maquina" : "abcd",
  "machine_id" : "123456",
  "update_d_time" : "11/04/2020 12:08:02",
  "empresa" : "Big data TI",
  "point" : [
    {
      "sensor_name" : "xpto",
      "sensor_desc" : "temperatura",
      "sensor_id" : "123789",
      "point_id" : [
        {
          "sensor_name" : "xpto",
          "last_seen" : NumberLong("1586617092000"),
          "update_d_time" : "11/04/2020 12:04:04",
          "rpm" : 980,
          "aceleracao_x" : 37.885,
          "aceleracao_y" : 33.66,
          "aceleracao_z" : 4.545,
          "aceleracao_local" : 50.67809018698317,
          "frequencia" : 980,
          "amplitude_x" : 3.06,
          "amplitude_y" : 4.775,
          "amplitude_local" : 5.67135124992272,
          "constante" : 894564.7303782004,
          "amplitude_z" : 4.233440961650271,
          "angulo_mov" : 32.653279826931794,
          "temperature_sensor" : 25.38,
          "displacement_x" : 6.12,
          "displacement_y" : 9.55,
          "displacement_z" : 0,
          "phase_angle" : 210.41
        }
      ]
    }
  ]
}
```

Serão feitas transformações para um novo formato, em que cada point\_id é uma chave e terá uma lista onde serão separados os dados por timestamp. Após esse processo de extração e transformação, os dados para cada point id serão armazenados na collection 'point\_ids\_day' em seu 'formato bruto'.

O formato de armazenagem nessa collection será:

Point\_ids\_day

Dado bruto

```
'id_sensor': [
  {'timestamp': [{ 'sensor_name': 'sensor_xpto',
    'last_seen': 1586617092000,
    'update_d_time': '11/04/2020 12:04:04',
    'rpm': 980.0,
    'aceleracao_x': 37.885,
    'aceleracao_y': 33.66,
    'aceleracao_z': 4.545,
    'aceleracao_local': 50.67809018698317,
    'frequencia': 980.0,
    'amplitude_x': 3.06,
    'amplitude_y': 4.775,
    'amplitude_local': 5.67135124992272,
    'constante': 894564.7303782004,
    'amplitude_z': 4.233440961650271,
    'angulo_mov': 32.653279826931794,
    'temperature_sensor': 25.38,
    'displacement_x': 6.12,
    'displacement_y': 9.55,
    'displacement_z': 0.0,
    'phase_angle': 210.41}]},
  {timestamp2: [{dados pra esse horario}]}
]
```

Esse ETL será executado a cada 50 segundos.

Nessa collection só deve permanecer os dados referentes as últimas 24 horas a partir de uma data. Por exemplo:

se agora são 9 horas da manhã, na collection point\_ids\_day devem permanecer os dados que tem seus timestamps entre 9 horas de ontem e 9 de hoje.

Para garantir isso, será criado um processo que fica responsável por a cada 23 horas e 59 minutos apagar dados com timestamps maiores do que 24 horas nessa collection depois de os ter inserido na collection point\_ids\_tri.

**2º Etapa point\_ids\_tri – Prazo limite: 22/04/2020 às 18H**

filtro estiver entre >24horas e <90 dias = média por hora

Point\_ids\_tri

Média por hora

A cada 1 dia pega tudo que for maior que 24 horas calculando a média por hora.

Esse ETL extrairá os dados que tiverem seus timestamps maiores que 24 horas da 'collection\_point\_ids\_tri' e transformará esses dados brutos em medias por hora, por exemplo:

```
'point_id': [  
  {  
    'Hora zero do dia 5': {  
      'sensor_name': 'sensor_xpto',  
      'media rpm pra essa hora': '123',  
      'meida aceleração pra essa hora': '123'  
      etc..  
    }  
    'Hora 1 do dia 5': {  
      'sensor_name': 'sensor_xpto',  
      'media rpm pra essa hora': '123',  
      'meida aceleração pra essa hora': '123'  
      etc..  
    }  
  }  
]
```

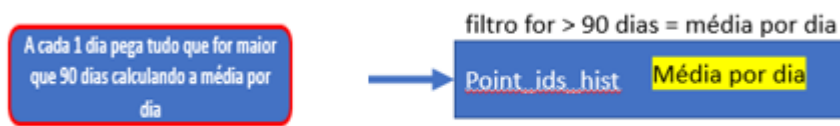
- Obs: os timestamps estarão no formato epoch

Após os dados serem transformados para essa estrutura eles serão carregados na collection point\_ids\_tri e nela permanecerão os dados que tiverem seus timestamps maiores do que as 24 horas de determinada data e menores do que 90 dias dessa data

( $24 < \text{timestamps} < 24 \times 90$ ). Esse ETL será executado a cada 23 horas e 59 minutos

Para garantir isso, será criado um processo que fica responsável por a cada 23 horas e 59 minutos apagar dados com timestamps maiores do que  $24 \times 90$  horas (90 dias) nessa collection depois de os ter inserido na collection point\_ids\_hist.

### 3ª Etapa point\_ids\_hist – 24/04/2020



O terceiro ETL extrairá os dados que tiverem seus timestamps maiores do que 90 dias da collection 'point\_ids\_tri', transformará para uma estrutura onde os timestamps serão divididos por dia e armazenarão as medias das variáveis por dia também, por exemplo:

```
'point_id': [  
  {  
    'dia 5 de abril': {  
      'sensor_name': 'sensor_xpto',  
      'media rpm pra essa hora': '123',  
      'meida aceleração pra essa hora': '123'  
      etc..  
    }  
    'dia 6 de abril': {  
      'sensor_name': 'sensor_xpto',  
      'media rpm pra essa hora': '123',  
      'meida aceleração pra essa hora': '123'  
      etc..  
    }  
  }  
]
```

- Obs: os timestamps estarão no formato epoch

Após os dados serem transformados para essa estrutura eles serão carregados na collection 'point\_ids\_hist' e nela permanecerão os dados que tiverem seus timestamps maiores do que as 24 x 90 horas (90 dias) de determinada data.

Esse ETL será executado a cada 23 horas e 59 minutos.

### Etapa 4 – Aplicação web – Data limite: 28/04/2020

Esse ETL atenderá a solicitação da aplicação web. A ideia é devolver dados com a estrutura proposta pelo Igor para um determinado intervalo de tempo.

### Estrutura proposta:

```
[
  {
    "delta_temperatura":{
      "alertas":[
        {
          "descricao":"warning",
          "cor":"red",
          "value": 29
        }
      ],
      "yAxis_descricao": "Temperatura (°C)",
      "linhas":[
        {
          "descricao":"linha 1",
          "cor":"blue",
          "x": [1.8, 1.8, 1.9, 10, 25]
        }
      ],
      "timestamp": [1510707600000, 1510707600000, 1510707600000, 1510707600000, 1510707600000, 1510707600000]
    },
    "ambient_temperatura":{
      "alertas":[],
      "yAxis_descricao": "Temperatura (°C)",
      "linhas": [
        {
          "descricao":"ambient_temperatura linha 1",
          "cor":"blue",
          "x": [1.8, 1.8, 1.9, 10, 25]
        }
      ],
      "timestamp": [1510707600000, 1510707600000, 1510707600000, 1510707600000, 1510707600000, 1510707600000]
    },
    "humidity":{
      "alertas":[],
      "yAxis_descricao": "humidity (%)",
      "linhas": [
        {
          "descricao":"humidity linha 1",
          "cor":"blue",
          "x": [1.8, 1.8, 1.9, 10]
        }
      ],
      "timestamp": [1510707600000, 1510707600000, 1510707600000, 1510707600000, 1510707600000, 1510707600000]
    }
  }
]
```

### Resumo:

- Quando a solicitação contiver um intervalo de tempo com a dimensão 'dia' os dados serão extraídos da 'point\_ids\_day', transformados para a estrutura solicitada carregados num json e devolvidos via api - com dados de medição por minuto.
- Quando a solicitação contiver um intervalo de tempo com a dimensão 'hora >24' ou 'dias' (**24 < timestamp < 24 h x 90**) os dados serão extraídos da 'point\_ids\_tri', transformados para a estrutura solicitada, carregados num json e devolvidos via api.
- Quando a solicitação contiver um intervalo de tempo com a dimensão 'meses' ou 'ano' (**timestamp > 90 x 24**) os dados serão extraídos da point\_ids\_hist, transformados para a estrutura solicitada, carregados num json e devolvidos via api.