**PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS  
NÚCLEO DE EDUCAÇÃO A DISTÂNCIA**

**Pós-graduação *Lato Sensu* em Ciência de Dados e Big Data**

**Romulos Machado**

**Phillip Furtado**

**Análise de dados de temperatura ambiente versus logs de alerta em UM Data Center**

Belo Horizonte

2020

**Romulos Machado**

**Phillip Furtado**

**Análise de dados de temperatura ambiente versus logs de alerta em UM Data Center**

Trabalho de Conclusão de Curso apresentado ao Curso de Especialização em Ciência de Dados e Big Data como requisito parcial à obtenção do título de especialista.

Belo Horizonte

2020

**SUMÁRIO**

[1. Introdução 4](#__RefHeading___Toc633_3942287182)

[1.1. Contextualização 4](#__RefHeading___Toc635_3942287182)

[1.2. O problema proposto 4](#__RefHeading___Toc637_3942287182)

[2. Coleta de Dados 6](#__RefHeading___Toc639_3942287182)

[3. Processamento/Tratamento de Dados 7](#__RefHeading___Toc641_3942287182)

[4. Análise e Exploração dos Dados 11](#__RefHeading___Toc643_3942287182)

[5. Criação de Modelos de Machine Learning 11](#__RefHeading___Toc645_3942287182)

[6. Apresentação dos Resultados 11](#__RefHeading___Toc647_3942287182)

[7. Links 12](#__RefHeading___Toc649_3942287182)

[REFERÊNCIAS 13](#__RefHeading___Toc651_3942287182)

# 1. Introdução

## 1.1. Contextualização

Para que data centers e seus respectivos equipamentos (servidores, grandes unidades de armazenamento e ativos de rede) consigam atender à necessidade dos seus usuários, é imperativo que permaneçam em funcionamento constante 24 horas por dia, 7 dias por semana. Consequentemente, assim como a maioria dos equipamentos que trabalham com processamento de informações, a temperatura nos data centers está diretamente relacionada com o bom funcionamento dos servidores, sendo fundamental que eles não superaqueçam durante a execução de suas atividades. Pois, caso isso aconteça, estes equipamentos correm o risco de se auto desligarem ou queimarem, o que pode causar uma parada não programada do data center, consequentemente a interrupção dos serviços por ele provido. Com isso, se faz necessário uma análise estatística e preditiva dos eventos de elevação da temperatura de data centers e suas consequências no funcionamento dos equipamentos.

## 1.2. O problema proposto

No que concerne a alta disponibilidade de serviços, identificar se registros de eventos de alta temperatura dentro de um data center estão ou não relacionados a danos registrados através de logs, a partir de dados coletados sobre o próprio ambiente e usando conceitos de ciências de dados para analisá-los, ajuda não somente a entender a extensão de problemas em equipamentos, como também a definir sistematicamente que determinadas condições estabelecidas, possivelmente irão gerar impactos negativados decorrentes e, por isso, deve ser evitados ao máximo.

Para desenvolvedor este trabalho, foram utilizados dados de temperatura de ambiente registrados por sensores internos de 30 servidores Dell, de modelo R720, de um determinado data center de Manaus, entre os anos de 2014 a 2020, com o objetivo de compará-los com dados de problemas ou erros de hardware apresentados dentro desse mesmo período. Onde esta comparação determinará se existe correlação entre os eventos de superaquecimento do data center e os problemas registrados nos equipamentos, dentro de um mesmo intervalo de tempo.

# 2. Coleta de Dados

A nossa base de dados consistiu em escolher 30 servidores todos de mesmas especificações e características de hardware e, partir deles, estudar dois tipos de logs de cada um desses servidores, que foram armazenados ao longo de aproximadamente 04 anos, através de um dispositivo chamado de Integrated Dell Remote Access Controller (IDRAC). Esse componente também instalado dentro de cada um desses servidores com total independência de acesso e isolado do sistema operacional, permitiu que ficassem registrados vários alertas relacionados às mais diversas falhas relativas ao hardware dessa máquinas, assim como também registrou dados de temperatura externa a estes servidores, ao longo desses 04 anos

Sendo assim, como foi indicado no parágrafo anterior, tivemos a oportunidade de trabalhar com dois tipos de logs, para cada servidor:

***-*** ***Log de Alertas (log\_servidorXX.csv)***: contendo dados sobre alertas de falhas do equipamento, tendo a seguinte estrutura:

|  |  |  |
| --- | --- | --- |
| **Nome da coluna/atributo** | **Descrição** | **Tipo** |
| Severity | Indica qual a severidade do alerta (“Normal”, “Warning”, ”Critical”) | Texto |
| Date | Indica a data e horário do alerta | Data/horário |
| Description | Indica a mensagem de alerta | Texto |

***- Log de Temperaturas (servidorXX\_inlettemp.csv)***: contendo dados de temperaturas registrados de hora em hora dos equipamentos, tendo a seguinte estrutura:

|  |  |  |
| --- | --- | --- |
| **Nome da coluna/atributo** | **Descrição** | **Tipo** |
| Average | Indica a temperatura média em Celsius, dentro do intervalo de tempo. | Numérico |
| Peak | Indica o maior pico de temperatura em Celsius, dentro do intervalo de tempo | Numérico |
| Time | Indica a data e hora a qual foi feito a captura de temperaturas, usada para os cálculos de pico e média | Data/horário |

# 3. Processamento/Tratamento de Dados

Para obtermos uma base mais consistente de dados e facilitar as associações, resolvemos adicionar o atributo “Server”, de tipo “Texto”, em cada um dos logs de alertas e também nos logs de temperaturas, para preenchê-los com as respectivas informações de nome dos servidores.

Isto facilitou o processo de criarmos um dataset único, contendo todos os registros de temperaturas referentes aos 30 servidores. Da mesma forma, foi criado um dataset único de registros de alertas de todos os respectivos servidores. Respectivamente, o dataset único de temperatura foi nomeado de “servidores-inlettemp.csv” (contendo 1541376 linhas, 4 colunas e sem valores nulos ou vazios) e o de logs de alertas foi chamado de “logs\_servidores.csv” (contendo 6368 linhas, 4 colunas e sem valores nulos ou vazios).

Usando a ferramenta Jupyter Notebook, e com isso facilitar as nossas interações com os datasets através do Python, assim tratamos e processamos os mesmos conforme a sequência descrita a seguir:

**3.1** Carregar arquivos originais de temperatura e log de servidores.

*import os*

*import numpy as np*

*import pandas as pd*

*temp = pd.read\_csv('datasets\_consolidados/servidores-inlettemp.csv')*

*logs = pd.read\_csv('datasets\_consolidados/logs\_servidores.csv')*

**3.2** Verificar a quantidade de linhas com valores nulos no dataset de temperaturas

*temp.isnull().sum()*

**3.3** Verificar a quantidade de linhas com valores nulos no dataset de logs de alertas

*logs.isnull().sum()*

**3.4** Remover as não-ocorrências de temperatura no dataset de temperaturas, pois evidencia que o servidor estava desligado e isso é caracterizado quando o atributo Average indica o valor de -128. Esse tipo de registro de máquina desligada não são relevantes para o nosso estudo, pois não fica inviável comparar temperatura com qualquer outra variável.

*temp.drop(temp.index[temp['Average'] == -128], inplace = True)*

**3.5** Transformar o atributo Time em tipo Date and cria um atributo DateOnly para posteriores comparações entre bases. Esse novo atributo irá nos ajudar a indexar e juntar as bases.

*temp['Time']= pd.to\_datetime(temp['Time'], format="%a %b %d %H:%M:%S %Y")*

*temp['DateOnly'] = temp['Time'].dt.date*

**3.6** Ordenar a base de temperaturas em função do nome do servidor e do dia/hora.

temp.sort\_values(by=['Server', 'Time'])

**3.7** Agrupar as linhas em função do nome de servidor e data, além de sumarizar através do agrupamento os atributos Average e Peak com informações de Mínimo, Máximo, Média, Mediana, Variância e Desvio Padrão. Para isso, é criado um novo dataframe resultante chamado temp\_summary.

temp\_summary = temp.groupby(['Server','DateOnly'])[['Average','Peak']].agg(['min', 'max','mean','median','var','std']).reset\_index()

**3.8** Como a operação de groupby adiciona uma nova linha de índices, pra isso é necessário realizar um ajuste para se manter uma só linha de índice sem perda de identidade das colunas.

*temp\_summary.columns = ["\_".join(x) for x in temp\_summary.columns.ravel()]*

*temp\_summary.rename(columns={'Server\_':'Server','DateOnly\_':'DateOnly'},inplace=True)*

**3.9** Remover as ocorrências da base de logs aonde a data tem formato inválido pois indica um processo de reinicialização da máquina (tecnicamente chamado de System Boot) , após desligamento, e transforma o atributo Date em tipo Date.

*logs.drop(logs.index[logs['Date'] == 'System Boot'], inplace = True)*

*logs['Date']= pd.to\_datetime(logs['Date'], format="%a %b %d %Y %H:%M:%S")*

**3.10** Criar o atributo DateOnly e ordernar a base de logs em função do nome do servidor e o dia do registro.

*logs['DateOnly'] = logs['Date'].dt.date*

*logs.sort\_values(by=['Server', 'Date'])*

**3.11** Combinar as duas bases em uma só, usando como índice o nome do servidor e o dia do registro. Neste caso levou-se em consideração o fato de que nem todas as vezes que houve um registro de temperatura, necessariamente, houve um registro de alerta. Para isso, é criado um novo dataframe resultante chamado merged.

*merged = pd.merge(temp\_summary,logs, on=['Server', 'DateOnly'], how='left')*

**3.12** No novo dataframe gerado no passo anterior, substituir o valores vazios (ausência de registro de log) da coluna Severity com o valor NoneAlert, indicando que nessa data não houve registros de alerta. Posteriormente, isso irá nos ajudar a efetuar comparações entre dados de temperatura e severidade de alertas.

*merged.Severity.fillna("NoneAlert", inplace=True)*

**3.13** Adiciona o atributo categórico TempSala que indica se a sala de servidores estava quente, esquentando ou fria, baseado nos valores de Peak\_max.

*merged.loc[merged['Peak\_max'] <= 26, 'TempSala'] = 'SalaFria'*

*merged.loc[(merged['Peak\_max'] > 26) & (merged['Peak\_max'] <= 33), 'TempSala'] = 'SalaEsquentando'*

*merged.loc[merged['Peak\_max'] > 33, 'TempSala'] = 'SalaQuente'*

**3.14** O atributo Severity será útil ao executar alguns modelos como o PCA na base merged, mas a maioria dos modelos requer que atributos categóricos sejam transformados em inteiros. Para isso, utilizaremos a técnica de One-Hot Encoding, na qual as categorias se transformaram em colunas (variáveis) onde o número 1 representa o valor afirmativo e o 0 negativo.

*merged.Severity.unique()*

**3.15** Cria-se então um dataset com os valores possíveis e utilizamos a função pd.get\_dummies() para gerar as colunas numéricas.

*df = pd.DataFrame({'Severity': ['NoneAlert','Normal', 'Critical', 'Warning']})*

*merged = pd.concat([merged,pd.get\_dummies(merged['Severity'], prefix='Severity')],axis=1)*

**3.16** Semelhante ao tratamento dado ao atributo Severity, também fizemos uso do mesmo processo no atributo TempSala.

*merged.TempSala.unique()*

*df = pd.DataFrame({'TempSala': ['SalaFria','SalaEsquentando','SalaQuente']})*

*merged = pd.concat([merged,pd.get\_dummies(merged['TempSala'], prefix='TempSala')],axis=1)*

**3.17** Com estes passos todos executados com sucesso, exportamos as bases para arquivos do tipo CSV, os quais serão utilizados no próximo tópico (merged.csv) deste trabalho e também usados como evidências de resultados de nosso tratamento de dados (temperatura.csv e logs.csv).

*temp.to\_csv(r'datasets\_consolidados/temperatura.csv', index = False)*

*logs.to\_csv(r'datasets\_consolidados/logs.csv', index = False)*

*merged.to\_csv(r'datasets\_consolidados/merged.csv', index = False)*

# 4. Análise e Exploração dos Dados

Nessa seção você deve mostrar como foi realizada a análise e exploração dos seus. Mostre as hipóteses levantadas durante essa etapa e os padrões e *insights* identificados.

# 5. Criação de Modelos de Machine Learning

Conforme o documento de instruções para o TCC, essa etapa não é obrigatória, mas é fortemente recomendada. Caso você crie modelos de *Machine Learn*ing em seu projeto, nessa seção você irá descrever as ferramentas utilizadas. Se você utilizou ferramentas visuais como Knime e Rapid Miner, coloque aqui um print do seu modelo. Caso você tenha escrito scripts em Python, por exemplo, coloque aqui o seu script. Explique as *features* utilizadas, justifique a escolha por determinado modelo, os parâmetros utilizados, etc.

# 6. Apresentação dos Resultados

Nessa seção você deve apresentar os resultados obtidos. Apresente gráficos, dahsboards, conte a sua história de forma bastante criativa. Aqui você pode utilizar os modelos de Canvas propostos por Dourard (clique [aqui](https://www.louisdorard.com/machine-learning-canvas)) ou por Vasandani (clique [aqui](https://towardsdatascience.com/a-data-science-workflow-canvas-to-kickstart-your-projects-db62556be4d0)).



# 7. Links

Aqui você deve disponibilizar os links para o vídeo com sua apresentação de 5 minutos e para o repositório contendo os dados utilizados no projeto, scripts criados, etc.

# REFERÊNCIAS

Um projeto de Ciência de Dados não requer revisão bibliográfica. Portanto, a inclusão das referências não é obrigatória. No entanto, caso você deseje incluir referências relacionadas às tecnologias ou às metodologias usadas em seu trabalho, relacione-as de acordo com o modelo a seguir.

SOBRENOME DO AUTOR, Nome do autor. **Título do livro ou artigo.** Cidade: Editora, ano.

SOBRENOME DO AUTOR, Nome do autor. **Título do livro ou artigo.** Cidade: Editora, ano.

SOBRENOME DO AUTOR, Nome do autor. **Título do livro ou artigo.** Cidade: Editora, ano.

SOBRENOME DO AUTOR, Nome do autor. **Título do livro ou artigo.** Cidade: Editora, ano.

SOBRENOME DO AUTOR, Nome do autor. **Título do livro ou artigo.** Cidade: Editora, ano.

SOBRENOME DO AUTOR, Nome do autor. **Título do livro ou artigo.** Cidade: Editora, ano.