



**FACULDADE SENAI DE TECNOLOGIA MECATRÔNICA**

**REDUÇÃO DO TEMPO DE RESPOSTA OPERACIONAL NA INDÚSTRIA 4.0 USANDO  
COMPUTAÇÃO EM NUVEM E BANCO DE DADOS ONLINE**

**REDUCTION OF OPERATIONAL RESPONSE TIME IN INDUSTRY 4.0 USING CLOUD  
COMPUTING AND ONLINE DATABASE**

Sylvia Cristina Kimura<sup>1, i</sup>  
Thiago Tadeu Amici<sup>2, ii</sup>  
José Roberto dos Santos<sup>3, iii</sup>  
Daniel Otávio Tambasco Bruno<sup>4, iv</sup>

**RESUMO**

A interconectividade trazida pela quarta revolução industrial possibilitará que a informação chegue até o operador e possa estar disponível *online*, sendo acessível em qualquer lugar e a qualquer hora. Este artigo mostra a viabilidade do uso do Banco de Dados Relacional (BDR) *online* e da Computação em Nuvem para realizar análises no auxílio da tomada de decisão para redução do tempo de resposta operacional como proposta de melhoria do sistema de automação de uma planta industrial. O BDR na nuvem será atualizado com os dados gerados pelo Sistema de Supervisão e Aquisição de Dados (SCADA) da Siemens® usando um *script* em *Python*, e as implementações de códigos na plataforma em nuvem da Amazon® Web Services (AWS) verificarão a existência de novos alarmes para notificar o operador via *e-mail*. Logo, serão desenvolvidos programas em *Python* para interagir com os recursos AWS Lambda e Amazon *Relational Database Service* (RDS) for MySQL na nuvem. Também serão apresentados uma análise de custos dos recursos usados na nuvem, uma estimativa do consumo de dados de uma planta industrial real e os tempos de processamento de dados na nuvem para análise do BDR. Esse cenário permitirá futuras implementações de análises no BDR na nuvem e melhorias de otimização dos processos industriais.

**ABSTRACT**

The interconnectivity brought by the fourth industrial revolution allows the information to reach the operator and it can be available online, being accessible anywhere and anytime. This article shows the feasibility of using an online Relational Database (RDB) and Cloud Computing

---

<sup>1</sup> Engenheira de Controle e Automação e Pós-graduanda em Indústria 4.0 na Faculdade SENAI de Tecnologia Mecatrônica. E-mail: syl\_kimura@yahoo.com.br

<sup>2</sup> Mestre em Automação e Controle de Processos da Faculdade SENAI de Tecnologia Mecatrônica. E-mail: thiago.amici@sp.senai.br

<sup>3</sup> Especialista em Segurança da Informação da Faculdade SENAI de Tecnologia Mecatrônica. E-mail: joseroberto@sp.senai.br

<sup>4</sup> Mestre em Engenharia da Informação. Professor da Faculdade SENAI de Tecnologia Mecatrônica. E-mail: daniel.bruno@sp.senai.br

to carry out analyzes to aid decision making to reduce operational response time as a proposal to improve the automation system of an industrial plant. The RDB will be updated with the data generated by the Siemens® Supervisory Control and Data Acquisition (SCADA) using a Python script and code implementations on the Amazon® Web Services (AWS) cloud platform will check the RDB for new alarms to notify the operator via email. Thus, there will be developed programs in Python to interact with the AWS Lambda and Amazon *Relational Database Service* (RDS) for MySQL in the cloud. A cost analysis of cloud resources, a data consumption estimate of a real industrial plant and the processing time of the code in the cloud will be presented. This scenario will allow future implementations of RDB analysis in the cloud and improvements to optimize industrial processes.

## 1 INTRODUÇÃO

A Tecnologia da Informação (TI) na área industrial vem trazendo significativas mudanças nos processos industriais com a digitalização dos dados e ter o *know-how* do gerenciamento desses dados é um dos pontos-chaves da quarta revolução industrial. De acordo com Greco et al. (2019), a transformação digital é necessária para as rápidas mudanças nos processos tecnológicos e fluxo contínuo de dados em massa, o que melhora o processo de tomada de decisão.

Aliado a essa transformação, é necessário ter um Banco de Dados (BD) que suporte toda essa estrutura de informações sendo geradas e requisitadas constantemente para fins de análises, previsões e digitalização da indústria. O uso do BD na nuvem permitirá o acesso *online* da informação por qualquer computador conectado à internet, a qualquer hora e em qualquer lugar. O uso dos recursos da nuvem trará economia de tempo na implantação, acesso e manutenção. De acordo com Shehri (2013), o banco de dados na nuvem se tornará a tecnologia mais adotada para armazenar grandes quantidades de dados pelas empresas, além de fornecer escalabilidade nas horas de pico, mantendo a performance e a Computação em Nuvem é bastante eficiente para recuperar informações, caso ocorra algum dano no banco de dados.

Ferramentas de análises de dados na nuvem poderão ser implementadas para auxiliar a tomada de decisão e redução do tempo de resposta operacional da planta industrial. Os paradigmas da quarta revolução industrial vão ser aplicados nesse contexto para que informações importantes de alarmes cheguem até o operador onde quer que ele esteja na planta industrial. “Um alarme é um sinal sonoro e/ou visual que indica ao operador uma falha no equipamento, desvio no processo ou condição anormal que requer uma ação.” (ANSI/ISA-18.2, 2016, apud ROCHA, 2016, p. 7).

O objetivo deste artigo é demonstrar a implementação do *backup* em tempo real no BDR da AWS, denominado Amazon RDS for MySQL, com os dados gerados pelo SCADA da Siemens®, e de um código em *Python* que será executado no AWS Lambda, na nuvem, para analisar o BDR para verificar novos alarmes e notificar o operador via *e-mail*. Também serão apresentados uma estimativa do consumo dos dados trafegados para a nuvem baseado em uma planta real de automação, dos tempos de processamento de dados na nuvem e dos custos dos recursos usados na AWS. Ou seja, na prática foram desenvolvidos um programa em *Python* para atualizar o AWS RDS for MySQL, o recurso AWS Lambda para executar os códigos em *Python* e um sistema SCADA desenvolvido no *Totally Integrated Automation* (TIA) Portal da Siemens® com WinCC RT Advanced para gerar alarmes, dados e testar a solução.

É importante ressaltar que a indústria esteja alinhada aos conceitos do *Lean* e Seis Sigma para reduzir ao máximo a quantidade de alarmes que são causados por falta de manutenção, e somente informar o que é importante. Segundo a *Engineering, Equipment and Material Users Association* (EEMUA) em sua publicação 191 (2007), um alarme a cada 5 minutos é gerenciável e um alarme a cada 10 minutos é aceitável (ALVES, 2013, p. 27; HONEYWELL®, 2011, p. 10). Os conceitos da melhoria contínua do *Lean* e do Seis Sigma de evitar o desperdício, contribuem nesse aspecto para manter o sistema de alarmes confiável.

O TIA Portal possui a funcionalidade de enviar alarmes por *e-mail*, além de existir empresas que desenvolvem *softwares* de gerenciamento de alarmes com notificação via celular, porém, a solução apresentada nesse artigo enfatiza o uso do BDR na nuvem e posteriores implementações de análises destes dados usando computação e recursos da nuvem, como notificar o operador em tempo real. Além de poder ser adaptada para qualquer sistema de automação, há a versatilidade de enviar quaisquer outras informações relativas ao alarme originado para orientação do operador. Também há o intuito de avaliar os tempos de respostas das análises feitas no BDR e da Computação em Nuvem para o estudo de viabilidade.

Esse artigo foi dividido em mais quatro seções, onde a segunda seção mostra os conceitos teóricos para fundamentação do artigo, a terceira seção mostra o desenvolvimento prático com a descrição dos programas e testes realizados, a quarta seção mostra os resultados obtidos, estimativa de custos e análises dos resultados, e na quinta seção são apresentadas as considerações finais.

## 2 FUNDAMENTAÇÃO TEÓRICA

### 2.1 Computação em Nuvem

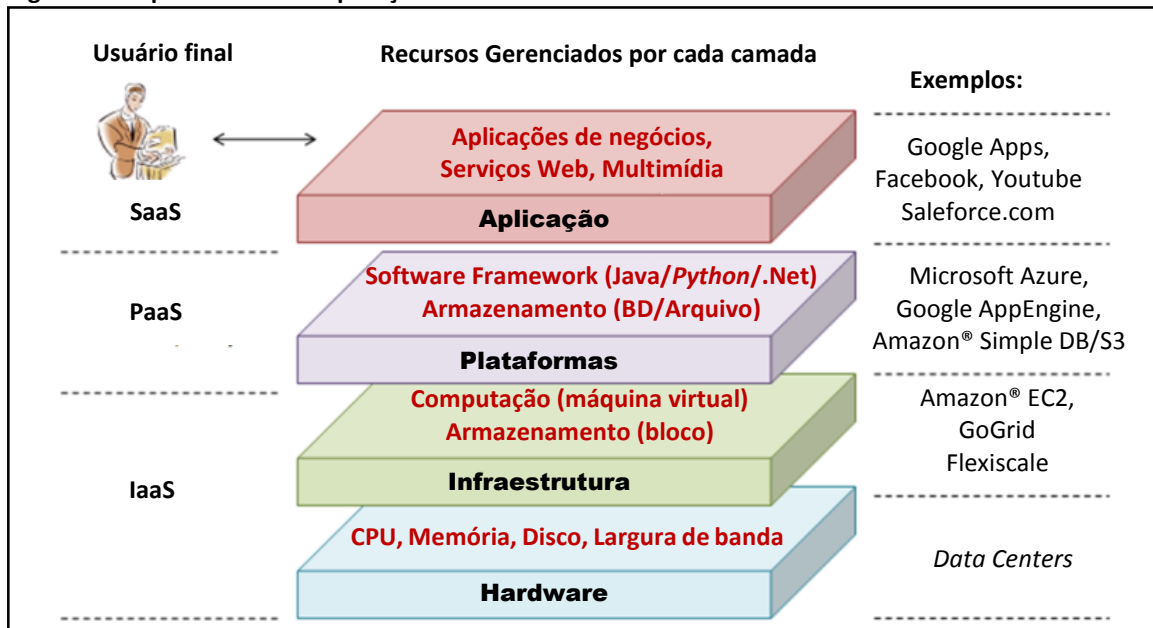
A indústria vem utilizando recursos de digitalização, análises dos dados, integração dos sistemas e de tecnologias de Computação em Nuvem para melhoria e redução do tempo dos processos industriais. O conceito da Indústria 4.0 foi introduzido pelo Governo Alemão, com a proposta de digitalizar a indústria. A Indústria 4.0 se baseia nos seguintes paradigmas: interoperabilidade, virtualização, descentralização, modularidade, reconfigurabilidade, tempo real e orientação a serviços (KOZÁK et al., 2018). E possui nove tecnologias habilitadoras ou digitais: Robótica Autônoma, Simulações, Manufatura Aditiva, Integração de Sistemas Vertical e Horizontal, Computação em Nuvem, Internet das Coisas Industrial (IIoT), Cibersegurança, Realidade Aumentada e *Big Data* (ALCÁCER e CRUZ-MACHADO, 2019).

A Computação em Nuvem fornece recursos computacionais, como processamento e armazenamento, que podem ser alugados e liberados para os usuários através da Internet e sob demanda (ZHANG et al., 2010). “A nuvem recebe esse nome justamente pela facilidade de acesso e uso, a qual permite que o programa que será utilizado não precise ser instalado no computador local” (IBM®, 2021). Seus principais benefícios são: a redução de custos de aquisição de *hardware* e *software*, e de atividades relativas à manutenção e disponibilidade dos servidores; a capacidade de dimensionamento elástico, com fornecimento da quantidade adequada dos recursos; a segurança e confiabilidade, com espelhamento dos dados em diversos sites redundantes e o uso de tecnologias com proteção aos dados (MICROSOFT®, 2020).

A arquitetura do ambiente da Computação em Nuvem pode ser dividida em quatro camadas, conforme mostra a figura 1. A camada *Hardware* gerencia os recursos físicos da nuvem, como servidores físicos, roteadores, switches e sistemas de refrigeração e energia. Em

geral é implementada nos *Data Centers*. A camada de Infraestrutura, ou virtualização, gera uma fonte de armazenamento e recursos de computação. A camada de Plataformas consiste nos sistemas operacionais e aplicações de *framework*. Nesta camada os usuários podem desenvolver, testar, fornecer e gerenciar aplicativos de *software*. E a camada de Aplicação consiste nas aplicações que são executadas na infraestrutura da nuvem. Os serviços oferecidos na nuvem podem ser agrupados em três categorias: Infraestrutura como Serviço (IaaS), formada pela camada de Hardware e de Infraestrutura; Plataforma como Serviço (PaaS), formada pela camada de Plataformas; e *Software* como Serviço (SaaS), formada pela camada de Aplicação. (ALCÁCER e CRUZ-MACHADO, 2019; MICROSOFT®, 2020; ZHANG et al., 2010).

Figura 1 - Arquitetura da Computação em Nuvem



Fonte: Adaptado pela autora de ZHANG et al. (2010).

Existem três tipos de nuvem: a nuvem pública, a privada e a híbrida. Na nuvem pública, os recursos computacionais de *hardware*, além dos custos de manutenção, são compartilhados entre os usuários, porém, com seguranças e regras no controle de acesso, impedindo que uma conta afete os outros usuários. Nesta nuvem, estes recursos pertencem a uma empresa terceirizada. A nuvem privada é usada por uma única empresa e os recursos computacionais lhe pertencem. Logo, o gestor de TI possui maior controle sobre eles. E a nuvem híbrida é uma mistura entre as duas, permitindo maior flexibilidade, com a movimentação dos dados e aplicativos entre ambas as nuvens (CENTRALSERVER, 2020; MICROSOFT®, 2020).

## 2.2 Banco de Dados

O banco de dados relacional utiliza a linguagem de busca estruturada (SQL) e consiste em uma ou mais tabelas usadas para representar os dados e as relações, que são compostas de linhas (tuplas) e colunas (atributos), e garantem segurança nas transações por apresentarem a conformidade ACID (atomicidade, consistência, isolamento e durabilidade) (FARIAS, 2014). Hammes et al. (2014) afirmam que a vantagem do banco de dados relacional

é a consistência dos dados garantida pelo ACID, pois a consistência dos dados em implementações na nuvem é fundamental, enquanto o BD *Not Only SQL* (NoSQL) segue o paradigma BASE (basicamente disponível, *soft state* e consistência eventual) e são usados quando a escalabilidade, disponibilidade e simplicidade são fundamentais, o que não garante uma atualização consistente de dados em um determinado instante.

Costa (2019) descreve os dois tipos de banco de dados, o relacional e o não-relacional, e faz uma relação entre vários autores que comparam esses BD e quais foram as metodologias usadas.

O BD relacional não oferece uma estrutura flexível e possui escalabilidade vertical, o que consiste em aumentar o poder de processamento e armazenamento das máquinas, enquanto o BD NoSQL permite maior flexibilidade dos dados, atendendo grandes volumes de dados semiestruturados ou não-estruturados que necessitam de alta disponibilidade e escalabilidade horizontal, o que é o processo de dividir o banco de dados em vários servidores (FARIAS, 2014; LÓSCIO et al., 2011).

Segundo Curino et al. (2011), o banco de dados como serviço ("*database-as-a-service*" - DBBaaS) é atrativo por duas razões: primeiro, devido às economias em escala, pois os custos incorridos provavelmente serão menores, já que os usuários estão pagando por uma parte do serviço e não por todo o *hardware*; e segundo, os custos incorridos do DBBaaS bem projetado serão proporcionais ao uso real do pagamento por uso, tanto das licenças de *software* quanto aos custos administrativos.

Como desvantagens apontada por Shehri (2013) sobre o BD na nuvem, é que não se tem controle sobre o servidor, nem dos *softwares* instalados nesses computadores; a dificuldade de se transferir todo o BD para um computador; a dificuldade de trocar de fornecedores de serviços de BD, pois diferenciam nos métodos e técnicas de armazenamento; e como o dado é trafegado pela internet, se o servidor ficar indisponível, poderá haver perdas caso a informação não seja acessada quando requisitada.

Logo, dentre os vários aspectos expostos, nesse artigo será usado um BD relacional na nuvem, o Amazon RDS for MySQL, pois os dados tratados são estruturados em tabelas e a consistência dos dados é relevante.

## 2.3 Plataforma em nuvem – AWS

Atualmente existem várias plataformas na nuvem que oferecem serviços de infraestrutura, plataforma e *software*, como a Amazon® com a AWS, a IBM® com o Watson, a Microsoft® com Azure, entre outros.

Conforme a Amazon® (2021a):

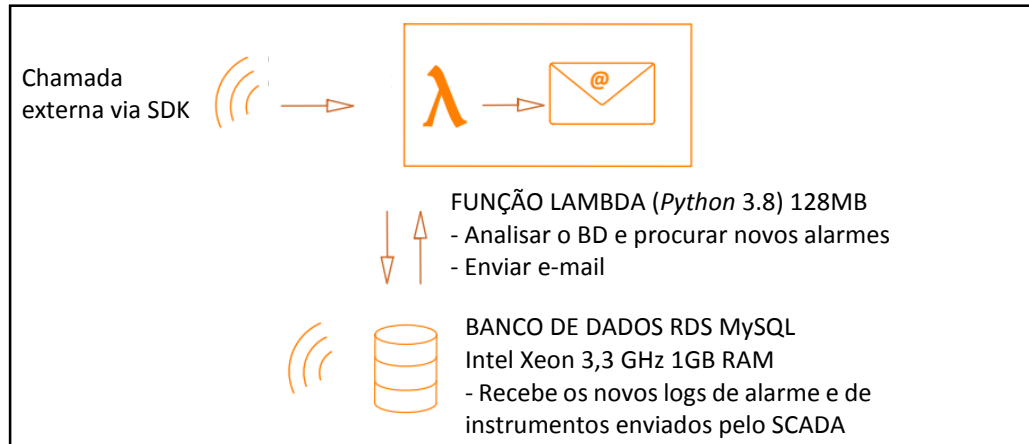
A Amazon® Web Services (AWS) é uma plataforma de nuvem [...] com serviços de computação, armazenamento, banco de dados, [...] machine learning, inteligência artificial, data lakes, análises e internet das coisas. [...] é compatível com 90 normas de segurança e certificações de conformidade, e todos os 117 serviços da AWS que armazenam dados de clientes oferecem criptografia de dados. [...] tem 77 zonas de disponibilidade em 24 regiões geográficas por todo o mundo. (AMAZON®, 2021a).

Serão usados os seguintes recursos da plataforma da Amazon®: Amazon RDS for MySQL e a função Lambda com *Python* 3.8, os quais não necessitam de servidor para funcionar, ou seja, não é necessário criar uma instância do Windows ou Linux, por exemplo, para executar esses serviços na nuvem. Basta configurá-los e utilizar suas funcionalidades. O

Amazon RDS for MySQL é um banco de dados relacional e a função Lambda permite a execução de códigos.

A figura 2 mostra os serviços e a relação dos recursos da AWS da Amazon® usados neste artigo.

**Figura 2 - Plataforma da AWS**



Fonte: Elaborado pela autora

### 2.3.1 AMAZON RDS for MySQL

O Amazon RDS for MySQL suporta as versões principais do banco de dados MySQL, como MySQL 8.0, 5.7, 5.6 e 5.5 e algumas versões secundárias.

Conforme a Amazon (2021b):

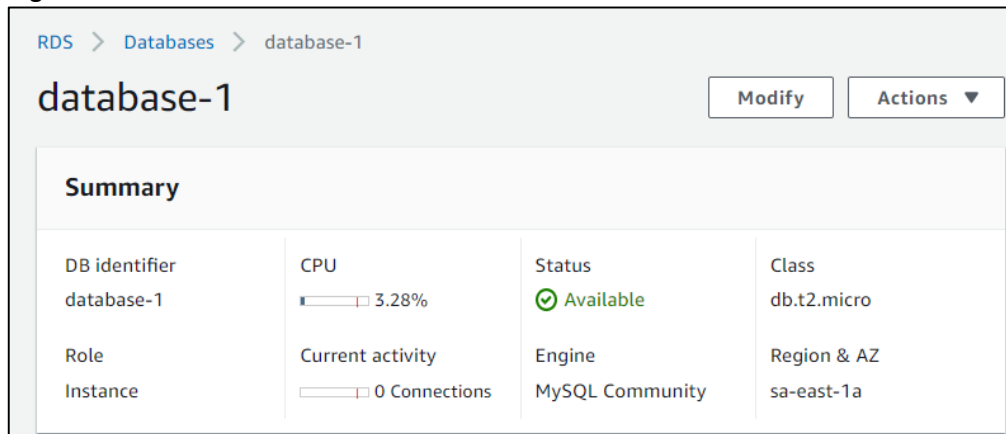
(...) é possível redimensionar a instância de banco de dados, autorizar conexões com a instância de banco de dados, criar e restaurar backups ou snapshots, criar Multi-AZ secundários, criar réplicas de leitura e monitorar o desempenho da instância de banco de dados. Você usa utilitários e aplicativos padrão do MySQL para armazenar e acessar os dados na instância de banco de dados. (AMAZON®, 2021b).

Neste artigo foi criada uma instância do tipo db.t2.micro com 20GB de armazenamento no Amazon RDS for MySQL na versão MySQL 8.0. Como o BD não vai ser acessado por um servidor na rede interna dentro da nuvem da Amazon®, deve ter acessibilidade pública habilitada para acesso externo e devem ser devidamente configuradas as regras de acesso de endereço de Protocolo de Internet (IP) em um “*Security group rules*” (regras do grupo de segurança) dentro da aba “*Connectivity & security*” (conectividade e segurança) da instância RDS com o IP de quem for ter acesso ao banco de dados.

A instância db.t2.micro possui um processador Intel Xeon de alta frequência com capacidade de intermitência, 1 vCPU (CPU virtual – *hyperthread* de um núcleo), processamento escalável de até 3,3GHz, 6 créditos de CPU/hora, 1 GB de memória RAM (*Random Access Memory*) e performance de rede em Gbps (gigabits por segundo) de baixa a moderada.

A figura 3 mostra a instância do Amazon RDS for MySQL criada no AWS com o nome de “database-1”.

Figura 3 – Instância do RDS no AWS



Fonte: Elaborado pela autora

### 2.3.2 AWS Lambda

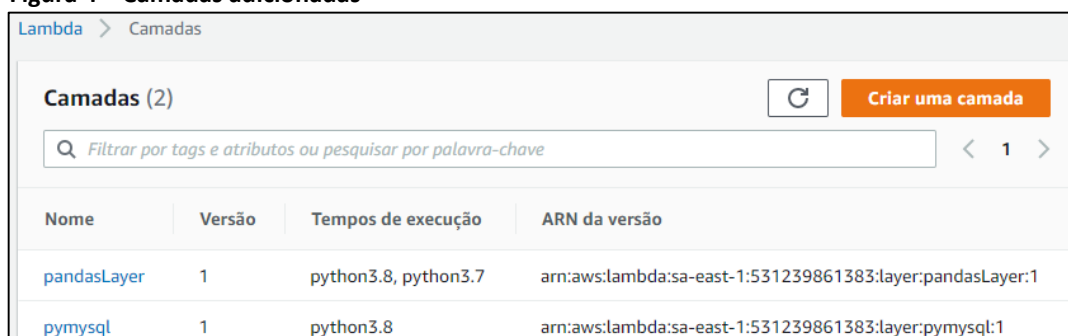
“Em vez da disponibilidade 24/7 comum nas camadas de pirâmide de nuvens IaaS, PaaS e SaaS, o AWS Lambda é orientado a eventos. O código é executado somente quando é necessário e consome zero recursos quando não é acionado.” (FEDAK, 2018).

Conforme definição da Amazon (2021c):

O AWS Lambda é um serviço de computação sem servidor que permite a execução de códigos, que podem ser chamados sob demanda ou com base em evento, como: requisições HTTP via Amazon® API (interface de programação de aplicativo) Gateway; Amazon® CLI (interface de linha de comando); alterações nos dados do Amazon® Simple Storage (S3) ou em tabelas do Dynamo DB; eventos temporizados programados no Amazon® Cloud Watch Events, cujo tempo mínimo é 1 minuto; ou usar chamadas de API usando o kit de desenvolvimento de software (SDK) da AWS. As linguagens de programação compatíveis são: .NET Core 3.1/2.1 (C#/PowerShell), Go 1.x, Java (11 ou 8), Node.js (14.x, 12.x ou 10.x), Python (3.8, 3.7, 3.6 ou 2.7), Ruby (2.7 ou 2.5). Os recursos computacionais são gerenciados automaticamente (AMAZON®, 2021c).

Para o desenvolvimento deste artigo, foram necessários instalar duas bibliotecas: a Pandas e a pymysql, pois elas não são instaladas por padrão. Cada biblioteca é instalada dentro de uma camada, onde é feito o *upload* da biblioteca. As bibliotecas instaladas ficam disponíveis para serem adicionadas dentro da função Lambda. A figura 4 mostra as camadas adicionadas.

Figura 4 – Camadas adicionadas

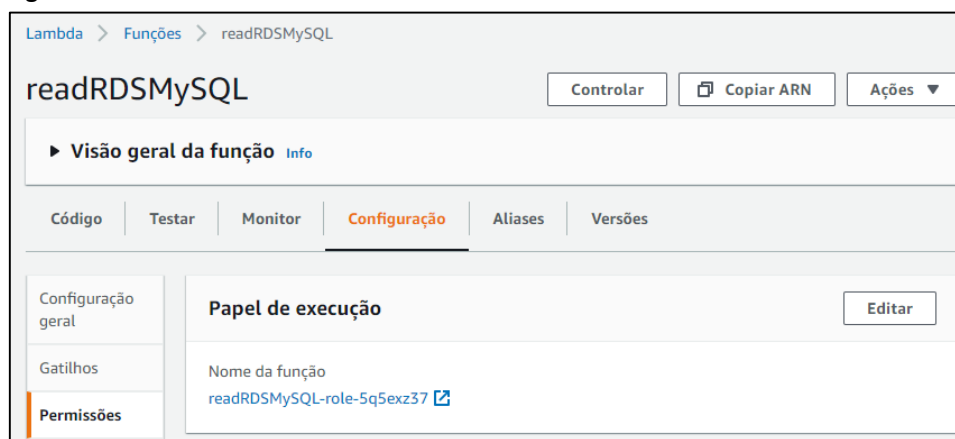


Fonte: Elaborado pela autora

Como gatilho para executar a função Lambda, foi feito uma chamada via SDK da AWS com *Python* no computador SCADA, para a função ser automaticamente executada na nuvem logo após a atualização do log de alarme no BD. Foi usado 128MB de RAM e tempo limite de execução de função de 1 minuto. É possível mudar a quantidade de memória RAM de 128MB para até 10.240MB e o tempo limite de execução para até 15 minutos.

É necessário adicionar a permissão para a função Lambda acessar o Amazon RDS for MySQL na aba “Configuração” da função Lambda criada. E no menu “Permissões”, em “Papel de execução”, é necessário selecionar o nome da função com a permissão configurada no serviço “*Identity and Access Management*” (IAM) da Amazon®. A figura 5 mostra a permissão configurada no AWS Lambda.

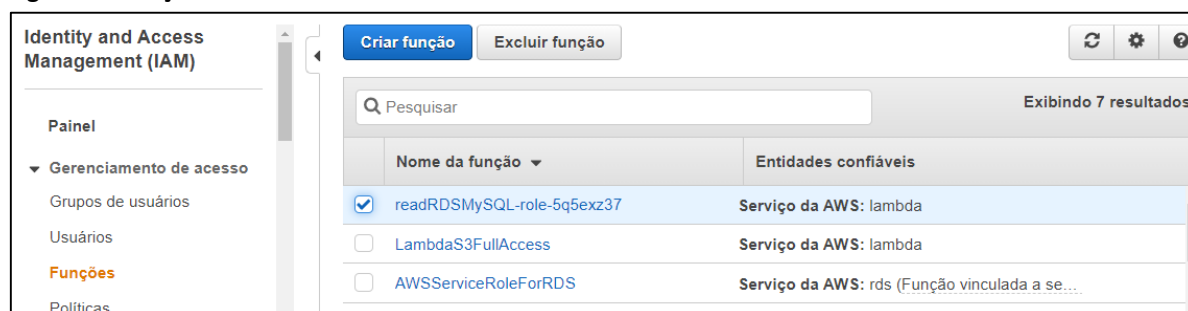
Figura 5 – Permissão do AWS Lambda



Fonte: Elaborado pela autora

A figura 6 mostra a função configurado no IAM, que deve ter a permissão “*AWSLambdaVPCLAccessExecutionRole*” adicionada para o AWS Lambda.

Figura 6 – Funções no IAM



Fonte: Elaborado pela autora

Para a função Lambda poder ser chamada via comando externo, é necessário criar uma ID da chave de acesso e a chave de acesso secreta, em “*My Security Credentials*” acessado pelo menu do usuário. Pode-se ter até duas chaves de acesso ativas ao mesmo tempo.

## 2.4 Sistema de automação

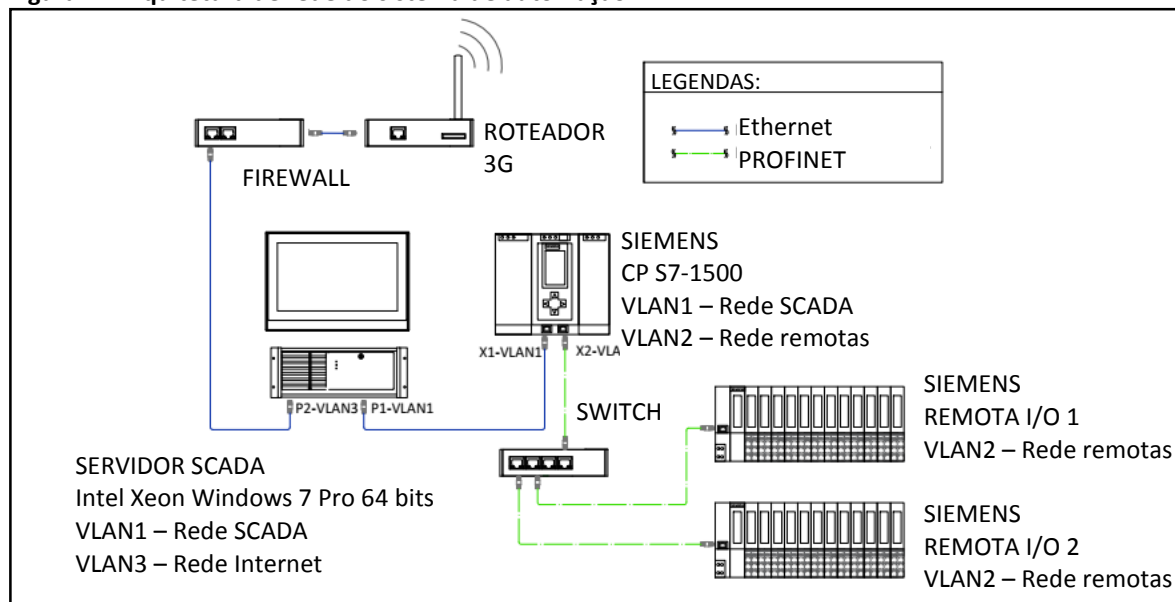
O sistema de automação possui um Controlador Programável (CP) da Siemens® S7-1500, que controla uma planta industrial com arquitetura de rede local, sem acesso à internet, conexão ponto a ponto entre CP e o computador SCADA, e sem integração com a rede



corporativa. O computador SCADA é um processador Intel® Xeon® com Windows 7 e precisa ter acesso à rede internet habilitado através de uma de suas portas de rede disponíveis. Para garantir a segurança dos dados no acesso à internet, deverá ser instalado um *firewall* entre o computador SCADA e o roteador de internet.

A figura 7 mostra a arquitetura de rede do sistema de automação.

**Figura 7 - Arquitetura de rede do sistema de automação**



Fonte: Elaborado pela autora

A programação do CP e do supervisor foi desenvolvido no TIA Portal V13. O supervisor gera arquivos com os valores separados por vírgulas (.CSV) dos históricos de cada instrumento, como o motor, alarmes e alertas. A tabela 1 mostra a quantidade de arquivos existente de cada tipo e a forma que é coletado o registro.

**Tabela 1 – Quantidade total de logs e tamanho do banco de dados**

Tipo de Registro	Quantidade de arquivos de cada tipo	Descrição do registro
Alarme	1	por evento
Alerta	1	por evento
Instrumentos analógicos	25	por período
Instrumentos digitais	6	mudança de estado
Motores	29	mudança de estado
Atuadores	8	mudança de estado

Fonte: Elaborado pela autora

O quadro 1 mostra um exemplo do log de alarmes que é gerado pelo CP, mas essa informação deve ser tratada antes de ser enviada para o operador, para ele receber apenas informações dos alarmes ativos (“StateAfter” = 1), a data, a hora (ambos são indicados no campo “TimeString”), e a descrição do alarme (indicado em “MsgText”). O log de alertas possui a mesma formatação de tabela que o log de alarmes. Os alertas são avisos que não são impeditivos para operação do sistema.

**Quadro 1 – Exemplo do Log de Alarmes**

Index	Time_ms	MsgProc	StateAfter	MsgClass	MsgNumber	Var1	Var2	Var3	Var4	Var5
0	44183918646,3079	2	1	1	1					
1	44195393650,6944	2	0	1	7					
2	44201792051,7477	2	1	1	2					

Index	Var6	Var7	Var8	TimeString	MsgText	PLC
0				"18/12/2020 22:02:51"	"Alarme 1 - Botao Emerg. atuado"	"HMI_Connection"
1				"30/12/2020 09:26:51"	"Alarme nivel muito alto tanque 1"	"HMI_Connection"
2				"05/01/2021 19:00:33"	"Alarme tanque 1 nivel baixo"	"HMI_Connection"

Fonte: Elaborado pela autora

O quadro 2 mostra um exemplo de log de instrumentos.

**Quadro 2– Exemplo do Log de instrumentos**

Index	VarName	TimeString	VarValue	Validity	Time_ms
0	"LIT-001"	"10/10/2019 04:41:57"	68	1	43748195799,7917
1	"LIT-001"	"10/10/2019 04:46:57"	68	1	43748199272,2107
2	"LIT-001"	"10/10/2019 04:51:57"	68	1	43748202744,6296

Fonte: Elaborado pela autora

A validade (*Validity*) igual a "1" no log significa que não houve perda de comunicação entre o supervisor e o CP, e o valor está atualizado. Quando há perda de comunicação, é mostrado a validade igual a "0" ou "2". O valor igual a zero é quando foi possível gravar o último valor válido do instrumento, mas não necessariamente é um valor atualizado. Portanto, ao gerar um gráfico dos dados relativo a um instrumento é importante informar se o valor nulo é perda de comunicação e desconsiderar esse valor, ou se realmente não houve leituras, ou o equipamento pode estar com defeito.

Os arquivos de logs no formato CSV, gerados pelo SCADA, serão consolidados no Amazon RDS for MySQL cada um em uma tabela, mantendo um *backup* atualizado dos dados do SCADA na nuvem. E com base nos log gerados, será feita uma estimativa do tráfego de dados para a nuvem.

### 3 DESENVOLVIMENTO PRÁTICO

Para o desenvolvimento prático e aplicação das ferramentas da Indústria 4.0 nesse artigo, foram desenvolvidos três programas em *Python* que serão explicados nos itens seguintes e um supervisor no TIA Portal com o WinCC RT Advanced da Siemens® para gerar os logs de alarmes e de instrumentos para realização dos testes.

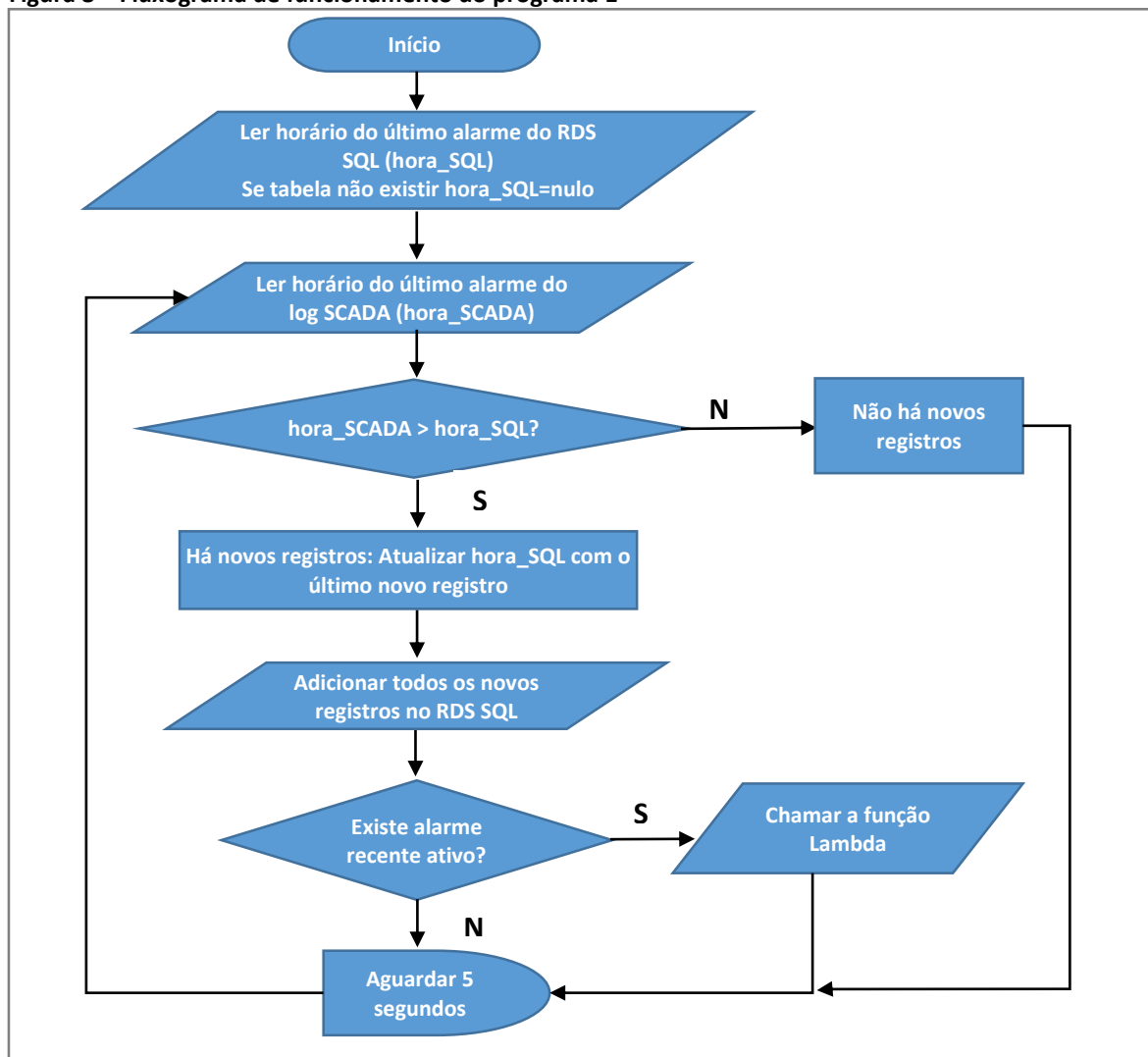
#### 3.1 Descrição dos programas desenvolvidos

##### 3.1.1 Programa 1 – SCADA: Sincronização dos logs de alarmes

Esse programa foi desenvolvido em *Python* 3.9, que é a versão mais atual para desenvolvedores, para ser executado no Servidor SCADA. Ele atualiza o Amazon RDS for MySQL com os logs de alarmes.

A figura 8 mostra o fluxograma de funcionamento desse programa.

Figura 8 – Fluxograma de funcionamento do programa 1



Fonte: Elaborado pela autora

A verificação de novos alarmes é feita a cada 5 segundos. E então, por comparação de horário entre o BDR na nuvem e do log no formato CSV, é verificado se há novos registros ou não. Quando há novos alarmes, ele manda executar a função AWS Lambda descrito no programa 3.

A tabela de dados gerada no Amazon RDS for MySQL é idêntica ao original, seguindo o exemplo do quadro 1. As principais bibliotecas usadas no programa são: Pandas, que faz a leitura do arquivo CSV e a análise dos dados do log; boto3, para interface com a AWS Função Lambda; e sqlalchemy, para conexão, leitura e gravação no Amazon RDS for MySQL.

Para fazer a chamada da função Lambda via SDK da AWS, é executado o código mostrado na figura 9, no qual é usada a biblioteca boto3 fornecida pela AWS. A função Lambda possui o nome “readRDSMySQL” e os parâmetros como região, chave de acesso e senha estão no arquivo “config.py” que é importado no início do *script*.

**Figura 9 – Código Python para chamada da função Lambda**

```
def invoke_lambda():
    region = config.region
    ACCESS_KEY = config.ACCESS_KEY
    SECRET_KEY = config.SECRET_KEY
    # Set Lambda Client with credentials
    boto3.setup_default_session(region_name=region)
    client = boto3.client('lambda', aws_access_key_id=ACCESS_KEY, aws_secret_access_key=SECRET_KEY)
    client.invoke(FunctionName='readRDMySQL', InvocationType='Event', LogType='None')
```

Fonte: Elaborado pela autora

### 3.1.2 Programa 2 – SCADA: Sincronização dos instrumentos

Programa desenvolvido em *Python* 3.9 para ser executado no Servidor SCADA. Ele atualiza o Amazon RDS for MySQL com os logs dos instrumentos, separando-os por tabelas, e usa a mesma instância do BDR para os logs de alarmes. Esse programa segue a mesma lógica do programa anterior, porém não faz a chamada da função Lambda, pois não há nenhum alarme nesses logs. Como os instrumentos não estão com periodicidade alta de gravação do histórico, pode-se ter um intervalo de verificação dos logs maior. Para os testes foi colocado um intervalo de 10 segundos.

A tabela de dados gerada no Amazon RDS for MySQL é idêntica ao original, seguindo o exemplo do quadro 2. As principais bibliotecas usadas no programa são: Pandas, que faz a leitura do arquivo CSV e a análise dos dados do log; e sqlalchemy, para conexão, leitura e gravação no Amazon RDS for MySQL.

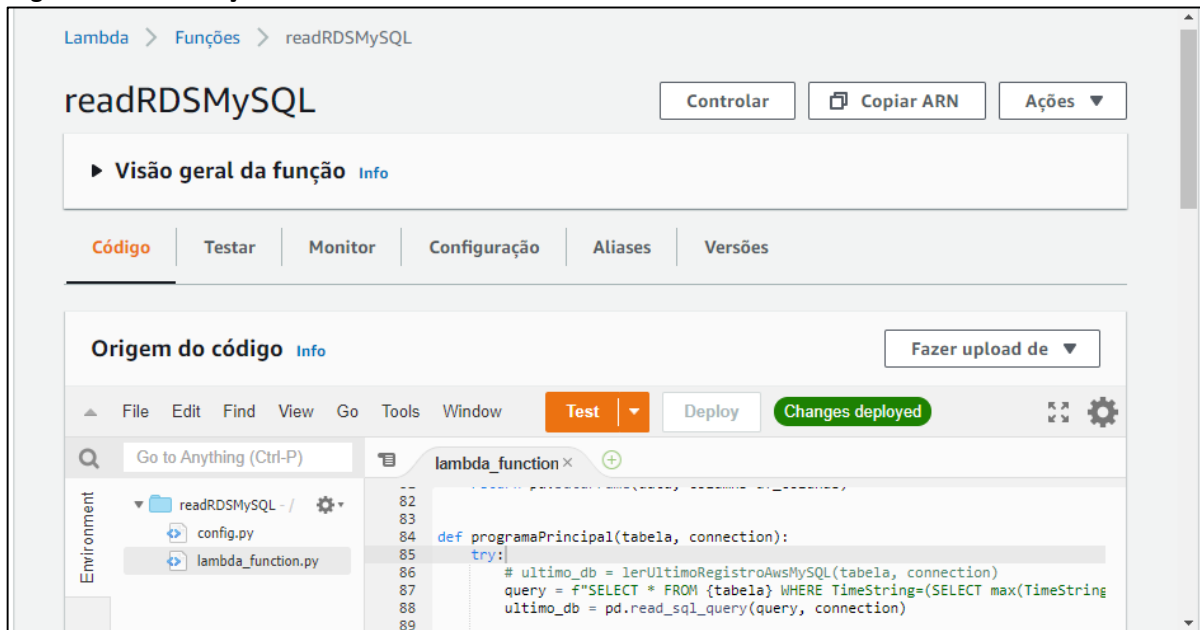
### 3.1.3 Programa 3 – AWS Lambda: Analisar BDR e enviar e-mail

Programa desenvolvido em *Python* 3.8 (versão mais atual na AWS) para executar uma função Lambda, que é chamado diretamente pelo programa 1. Quando executado, ele verifica na tabela de alarmes do Amazon RDS for MySQL se, no último minuto, ainda há algum alarme ativo que não foi reconhecido pelo operador e não tenha sido desativado. E então, manda esses alarmes ativos por *e-mail*. Caso este alarme já tenha sido reconhecido ou desativado, não há necessidade de alertar o operador.

As principais bibliotecas usadas no programa são: pymysql, para conexão e leitura dos dados no Amazon RDS for MySQL; Pandas, para análise dos dados dos logs; smtplib, para envio *dos e-mails*; e datetime, para operações nas variáveis de tempo.

A figura 10 mostra a tela da função Lambda na AWS criada com o nome de “readRDMySQL”.

Figura 10 – Tela função Lambda

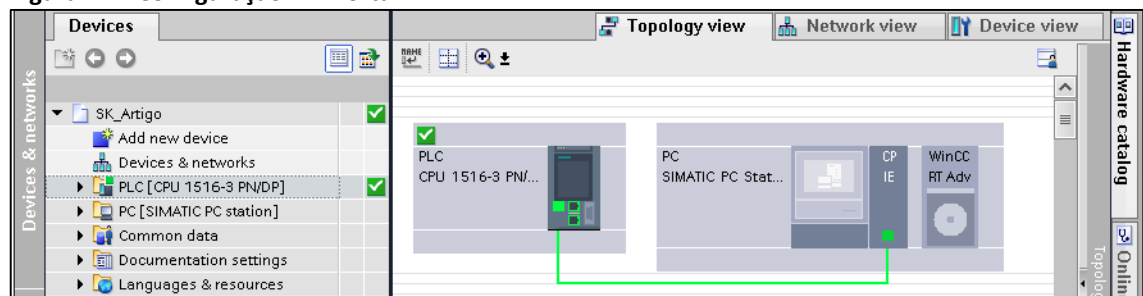


Fonte: Elaborado pela autora

### 3.1.4 Supervisório – Gerar alarmes e dados

No TIA Portal V13 foi configurado o CP S7-1500 com conexão a um computador, denominado “SIMATIC PC Station”, com o WinCC RT Advanced para desenvolver o programa supervisório, conforme mostra a figura 11. Foi usado o programa S7-PLCSIM para simular este CP.

Figura 11 – Configuração TIA Portal



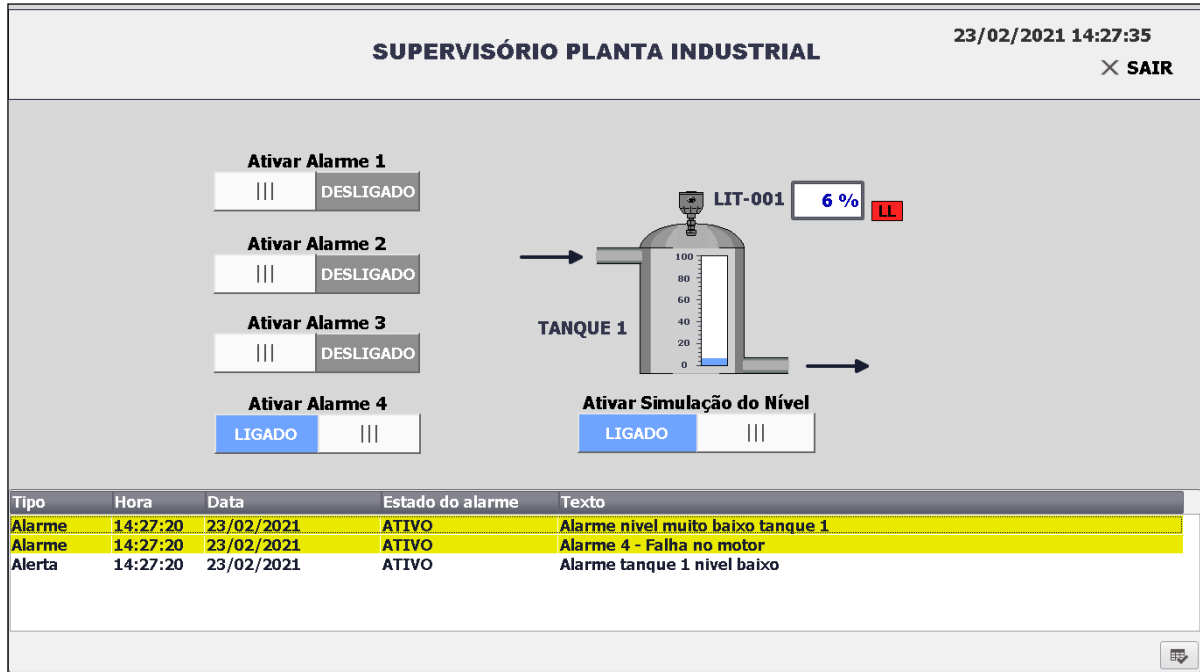
Fonte: Elaborado pela autora

No WinCC RT Advanced, em “HMI alarms” foram criados quatro alarmes que são ativados pelos interruptores, que mostram estado “Ligado” ou “Desligado”, e dois alarmes do transmissor de nível. Em “Historical data”, na aba “Data logs”, foi configurado a coleta do estado dos interruptores por mudança de estado, e a coleta, a cada 2 segundos, da leitura do transmissor de nível que está sendo simulado por uma lógica no CP. Na aba “Alarm logs” foi configurado a gravação dos logs dos alarmes. Os logs são escritos em arquivos individuais em formato CSV e são atualizados sempre que há um novo registro. O supervisório segue os conceitos da norma ANSI/ISA-101 (2015), que afirma que no projeto da Interface Homem-Máquina (IHM) deve-se levar em conta as deficiências de percepção de cores, sendo que os tipos mais comuns de daltonismo são: vermelho-verde, verde-amarelo e branco-ciano.

Como regra geral, as cores e animações devem ser usadas para enfatizar condições anormais e alarmes.

A figura 12 mostra a tela do supervisório desenvolvida, que foi usada durante os testes.

Figura 12 - Tela do supervisório de teste



Fonte: Elaborado pela autora

### 3.2 Testes realizados

Os testes das funcionalidades dos programas foram feitos usando um notebook de teste, marca Dell, com processador I7 e conexão estável à rede Internet, o qual ficava armazenado o arquivo de logs gerado pelo TIA Portal. O servidor de *e-mail*, utilizado para os testes, foi o Yahoo®.

O *software* MySQL Workbench 8.0 da Oracle® foi usado para visualizar as tabelas da instância do Amazon RDS for MySQL. No Workbench, as configurações de *hostname* e *port* são indicados, respectivamente em *endpoint* e *port*, na seção de “Conectividade e segurança” da instância do BDR do Amazon RDS for MySQL.

A figura 13 mostra a visualização dos registros dos logs de um instrumento no Amazon RDS for MySQL utilizando o Workbench.

Figura 13 – Visualização dos registros de um instrumento

	VarName	TimeString	VarValue	Validity	Time_ms
	LIT-001	2021-02-01 20:11:45	75,73112	1	44228841488,7963
	LIT-001	2021-02-01 20:11:48	72,31609	1	44228841522,419
	LIT-001	2021-02-01 20:11:50	68,76249	1	44228841545,7176
	LIT-001	2021-02-01 20:11:52	65,32781	1	44228841569,0394
	LIT-001	2021-02-01 20:11:54	61,84821	1	44228841592,3495
	LIT-001	2021-02-01 20:11:56	58,03604	1	44228841617,0602
	LIT-001	2021-02-01 20:11:58	54,65105	1	44228841640,3704

Fonte: Elaborado pela autora

A figura 14 mostra os logs de alarmes no Amazon RDS for MySQL utilizando o Workbench.

Figura 14 - Visualização dos registros de alarmes

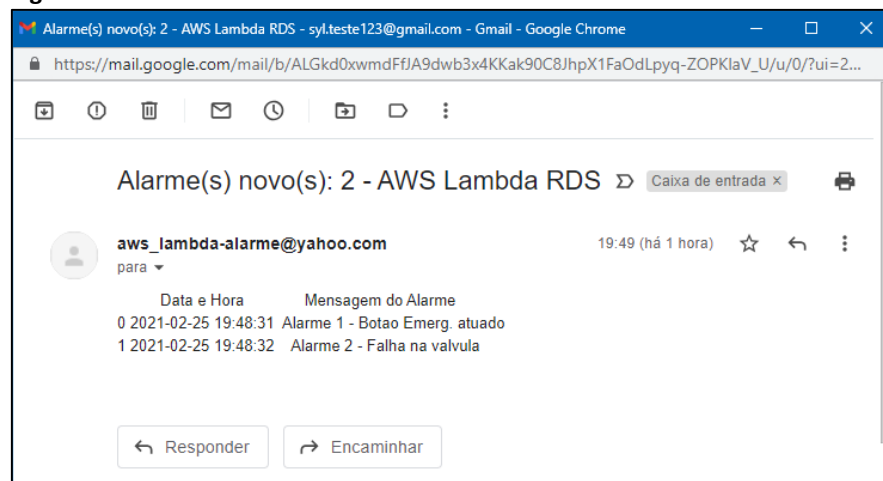
Time	MsgF	State	MsgI	Msg	Var1	Var2	Var3	Var4	Var5	Var6	Var7	Var8	TimeString	MsgText	PLC
44...	2	0	1	3	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	2021-02-25 19:48:30	Alarme 3 - Sobrecarga no motor	HM...
44...	2	0	1	4	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	2021-02-25 19:48:30	Alarme 4 - Falha no motor	HM...
44...	2	1	1	1	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	2021-02-25 19:48:31	Alarme 1 - Botao Emerg. atuado	HM...
44...	2	1	1	2	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	2021-02-25 19:48:32	Alarme 2 - Falha na valvula	HM...
44...	2	0	1	2	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	2021-02-25 19:50:06	Alarme 2 - Falha na valvula	HM...
44...	2	0	1	1	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	2021-02-25 19:50:07	Alarme 1 - Botao Emerg. atuado	HM...
44...	2	1	1	4	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	2021-02-25 19:50:08	Alarme 4 - Falha no motor	HM...
44...	2	1	1	2	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	2021-02-25 19:50:09	Alarme 2 - Falha na valvula	HM...
44...	2	0	1	2	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	2021-02-25 19:51:19	Alarme 2 - Falha na valvula	HM...
44...	2	0	1	4	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	2021-02-25 19:51:20	Alarme 4 - Falha no motor	HM...
44...	2	1	1	3	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	2021-02-25 19:51:30	Alarme 3 - Sobrecarga no motor	HM...
44...	2	1	1	4	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	2021-02-25 19:51:30	Alarme 4 - Falha no motor	HM...

Fonte: Elaborado pela autora

#### 4 RESULTADOS OBTIDOS

Os testes dos aplicativos desenvolvidos e do uso da plataforma em nuvem foram satisfatórios. Os *e-mails* recebidos possuíam as informações da data e hora, a descrição do alarme ativo e a quantidade de alarmes ativos gerados. A figura 15 mostra um exemplo do *e-mail* recebido com os alarmes. E o arquivo final, atualizado na nuvem, conteve todas as informações idênticas ao histórico do SCADA.

Figura 15 – E-mail de alarme recebido



Fonte: Elaborado pela autora

A tabela 2 mostra o resultado dos testes realizados com: o cálculo do tempo entre a notificação do alarme no SCADA e o recebimento do *e-mail*; o tempo de processamento da função Lambda para pesquisar o BDR e enviar os *e-mails*; a quantidade de memória usada para processamento; e a hora da chamada da função Lambda. Os resultados demonstrados na tabela 2 referem-se ao BDR praticamente vazio, e o tempo de resposta e de processamento foram relativamente curtos. Foram testados o tempo de processamento com um, dois e quatro alarmes ao mesmo tempo, com cinco repetições de cada.

A média do tempo de processamento foi de 3,8 segundos, com tempo máximo de 5,1 segundos. A média do tempo entre o alarme e recebimento do *e-mail* foi de 16 segundos. Não houve nenhuma taxa de erro durante os testes.

**Tabela 2 – Tempo de resposta da análise do BDR vazio e do recebimento do e-mail**

Data	Hora alarme (GMT-3)*	Hora recebimento e-mail (GMT-3)*	Qtd. de alarmes	Hora registrado na AWS Lambda (GMT-0)*	Duração Função Lambda [s]	Memória usada [MB]	Atraso [s]
19/02/21	23:26:31	23:26:46	1	2021-02-20T02:26:42.144Z	3,469	126	15
19/02/21	23:29:54	23:30:11	1	2021-02-20T02:30:05.991Z	3,888	126	17
19/02/21	23:31:18	23:31:32	1	2021-02-20T02:31:25.824Z	3,247	126	14
19/02/21	23:34:07	23:34:23	1	2021-02-20T02:34:18.842Z	2,601	126	16
19/02/21	23:39:12	23:39:30	1	2021-02-20T02:39:23.794Z	3,293	126	18
19/02/21	23:40:33	23:40:52	2	2021-02-20T02:40:48.490Z	4,149	126	19
19/02/21	23:42:31	23:42:39	2	2021-02-20T02:42:36.627Z	2,696	126	8
19/02/21	23:44:52	23:45:09	2	2021-02-20T02:45:04.378Z	5,088	126	17
19/02/21	23:46:39	23:46:52	2	2021-02-20T02:46:49.390Z	3,856	126	13
19/02/21	23:47:35	23:47:51	2	2021-02-20T02:47:46.044Z	3,692	126	16
19/02/21	23:48:45	23:49:03	4	2021-02-20T02:48:57.261Z	4,82	126	18
19/02/21	23:51:28	23:51:46	4	2021-02-20T02:51:43.441Z	4,996	126	18
19/02/21	23:53:07	23:53:20	4	2021-02-20T02:53:17.529Z	2,701	126	13
19/02/21	23:55:03	23:55:19	4	2021-02-20T02:55:15.187Z	5,047	126	16
19/02/21	23:56:23	23:56:38	4	2021-02-20T02:56:33.875Z	2,83	126	15

\* GMT - Tempo médio de Greenwich

Fonte: Elaborado pela autora

Para uma melhor avaliação desses tempos, foi colocado 2GB de dados no Amazon RDS for MySQL. A tabela de alarmes ficou com 597MB e com um total de 4.401.231 registros. A tabela 3 mostra o resultado do mesmo teste da tabela 2 com BDR iniciando com 2GB.

**Tabela 3 - Tempo de resposta da análise do BDR com 2GB e do recebimento do e-mail**

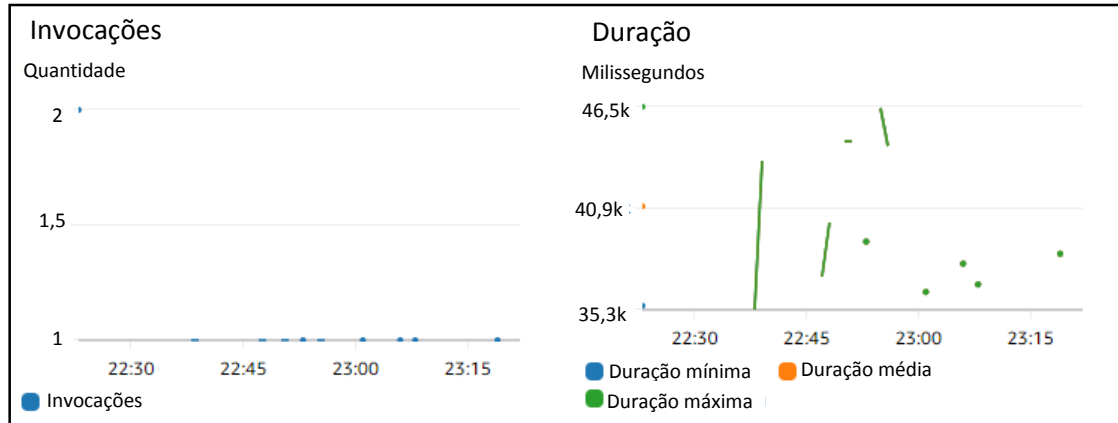
Data	Hora alarme (GMT-3)	Hora recebimento e-mail (GMT-3)	Qtd. de alarmes	Hora registrado na AWS Lambda (GMT-0)	Duração Função Lambda [s]	Memória usada [MB]	Atraso [min]
25/02/2021	19:15:20	19:16:10	1	2021-02-25T22:16:05.386Z	40,553	126	00:50
25/02/2021	19:19:37	19:20:29	1	2021-02-25T22:20:17.027Z	35,443	126	00:52
25/02/2021	19:20:25	19:22:30	1	2021-02-25T22:21:06.888Z	38,217	126	02:05
25/02/2021	19:23:45	19:26:05	1	2021-02-25T22:24:26.012Z	35,508	125	02:20
25/02/2021	19:38:03	19:38:54	1	2021-02-25T22:38:45.118Z	35,289	125	00:51
25/02/2021	19:39:17	19:40:08	2	2021-02-25T22:40:04.698Z	43,487	125	00:51
25/02/2021	19:47:19	19:48:14	2	2021-02-25T22:48:02.582Z	37,152	125	00:55
25/02/2021	19:48:31	19:49:25	2	2021-02-25T22:49:18.636Z	40,150	125	00:54
25/02/2021	19:50:08	19:51:04	2	2021-02-25T22:50:58.677Z	44,591	125	00:56
25/02/2021	19:51:30	19:52:26	2	2021-02-25T22:52:18.069Z	44,586	125	00:56
25/02/2021	19:53:38	19:54:32	4	2021-02-25T22:54:24.268Z	39,058	125	00:54
25/02/2021	19:54:59	19:55:59	4	2021-02-25T22:55:51.423Z	46,402	125	01:00
25/02/2021	19:56:43	19:57:39	4	2021-02-25T22:57:33.585Z	44,328	125	00:56
25/02/2021	20:06:45	20:07:38	4	2021-02-25T23:07:29.328Z	37,834	125	00:53
25/02/2021	20:08:15	20:09:03	4	2021-02-25T23:08:56.379Z	36,688	125	00:48

Fonte: Elaborado pela autora



A figura 16 mostra as métricas dos testes, realizados na tabela 3, referente a função Lambda que estão disponíveis na plataforma da AWS. Observa-se a quantidade de chamadas do programa (invocações) e a duração da sua execução em milissegundos.

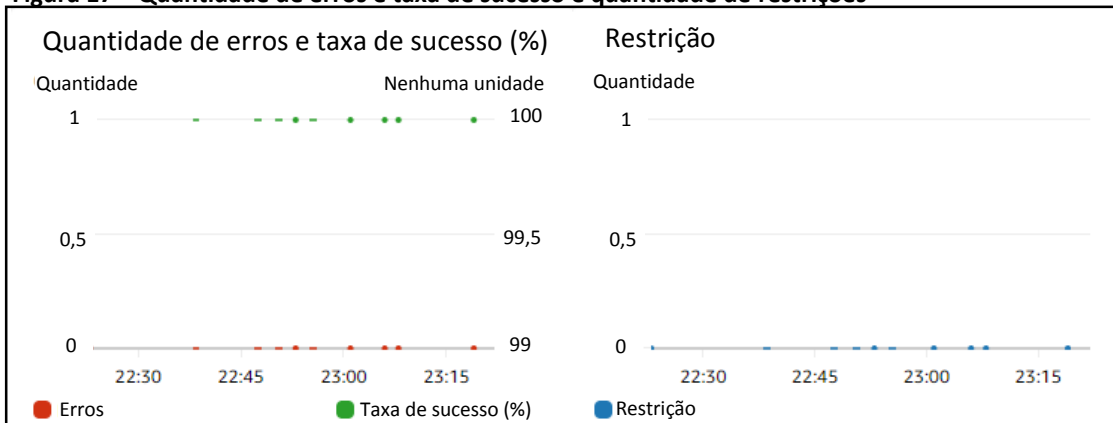
**Figura 16 – Quantidade de invocações e duração**



Fonte: Elaborado pela autora utilizando as métricas da AWS

A figura 17 mostra a quantidade de erros, acertos e quantidade de restrições. A restrição ocorre quando o AWS Lambda não invoca uma função por atingir o limite de simultaneidade da conta. Esse limite especifica o número de chamadas de função que podem ser executadas ao mesmo tempo.

**Figura 17 – Quantidade de erros e taxa de sucesso e quantidade de restrições**



Fonte: Elaborado pela autora utilizando as métricas da AWS

Observa-se que os tempos aumentaram com o BDR iniciando com 2GB, com média de 1 minuto e o máximo de 2,2 minutos de atraso para recebimento do e-mail. Já o tempo de processamento da função Lambda para analisar o BDR ficou em torno de 40 segundos.

#### 4.1 Análise do consumo de dados da planta real de automação

No SCADA, há a limitação de 500.000 registros por arquivo de log e os registros mais antigos são sobrescritos quando esse limite é atingido. Então para se estimar o BD cheio do servidor SCADA da planta real de automação descrito no item 2.4, foi feita uma manipulação dos arquivos de logs e preenchidos todos com 500.000 registros para calcular o BD cheio. Na tabela 4 é mostrado o tamanho desses arquivos cheios e quantos arquivos de cada tipo

existem no BD da planta de automação. Esse cálculo foi feito para a análise da viabilidade da Computação em Nuvem.

**Tabela 4 – Quantidade total de logs e tamanho do banco de dados**

Tipo de Registro	Tamanho de cada arquivo cheio [MB]	Quantidade de arquivos de cada registro	Totalização por tipo de registro [MB]
Alarme	55	1	55
Alerta	55	1	55
Instrumentos analógicos	28	25	700
Instrumentos digitais	28	6	168
Motores	28	29	812
Atuadores	28	8	224
Soma do tamanho total do banco de dados SCADA cheio			2.014 MB

Fonte: Elaborado pela autora

Portanto, o BD do SCADA cheio seria de 2 GB, resultando em um *upload* inicial de 2GB para o RDS for MySQL.

Para calcular o consumo dos dados trafegados para a nuvem por mês, foi feito o cálculo da quantidade média mensal de logs de cada tipo de registro da planta industrial do histórico existente de um ano, conforme mostra a tabela 5. Com esse valor da quantidade média mensal, foram gerados arquivos de logs com essa mesma quantidade para se verificar o tamanho do arquivo gerado, acrescido de 10%, para se ter uma estimativa do espaço ocupado por esses dados. Esse valor é mostrado na coluna “Tamanho do arquivo com a qtd. média de registros mensal” na tabela 5, que representaria uma estimativa do consumo de *upload* entre o SCADA e a nuvem por mês e do crescimento mensal do BD.

A quantidade total de alarmes e alertas indicada incluem os registros ativos e inativos.

**Tabela 5 – Estimativa do tamanho dos dados gerado por mês**

Tipo de Log	Quantidade média de registros por mês	Tamanho do arquivo com a qtd. média de registros mensal [KB]
Alarme	8850	2.000
Alerta	120	20
Instrumentos analógicos	555000	35.000
Instrumentos digitais	3600	200
Motores	31500	1.800
Atuadores	1860	200
Soma		39,220 MB

Fonte: Elaborado pela autora

Portanto, ter-se-ia um *upload* inicial de 2GB, referente ao BD cheio do SCADA para o RDS for MySQL, consumo de tráfego e crescimento do BD de aproximadamente 40 MB mensal.

Os instrumentos analógicos estão com periodicidade para gravar os registros, a cada 1 minuto, 5 minutos, 10 minutos, ou mais, dependendo da situação. Em geral, não possuem criticidade para atualizar por segundo, sendo que qualquer avaria entre os tempos de coleta são gerados alarmes. Caso contrário, haveria uma quantidade bem maior de registros por

mês, cerca de 64.800.000 logs por mês, gerados pelos 25 instrumentos analógicos com uma taxa de atualização de um segundo. Isto geraria uma estimativa de consumo de dados no total de 3,7GB por mês, implicando no crescimento mensal do BDR em aproximadamente 4 GB.

#### 4.2 Análise dos custos

A estimativa de custo dos serviços da AWS foi calculada baseado no comportamento mensal da planta industrial, conforme descrito no item 4.1, com crescimento do banco de dados mensal de 40MB. Os dados do quadro 3 foram obtidos da Calculadora de Custos da AWS Lambda (AMAZON, 2021d).

**Quadro 3 - Estimativa custos AWS Lambda**

<b>Conversões de unidades</b>		
Quantidade de memória alocada:	=	0,125 GB
128 MB x 0,0009765625 GB em um MB		
<b>Cálculo dos custos para região América do Sul (São Paulo)</b>		
11.160 requisições x 60.000 ms x 0.001 ms	=	669.600,00 processamento total (segundos)
0,125 GB x 669.600,00 segundos	=	83.700,00 processamento total (GB-s)
83.700,00 GB-s x 0,0000166667 USD	=	1,40 USD valor do processamento mensal
11.160 requisições x 0,0000002 USD	=	0,00 USD valor da requisição mensal
<b>Custos do Lambda mensal: 1,40 USD</b>		

Fonte: Elaborado pela autora

Como base de cálculos para a função Lambda, foi considerado 1 alarme a cada 5 minutos, equivalente a 12 requisições por hora, que são 8.928 requisições por mês. Porém, foi considerado um adicional de 25% que são 11.160 requisições por mês, equivalente a 1 alarme a cada 4 minutos ou 15 requisições por hora. O tempo de execução da função Lambda foi considerado 1 minuto, que é o tempo limite configurado.

Para estimar o custo do Amazon RDS for MySQL, foram considerados as seguintes configurações: região América do Sul (São Paulo), 1 instância RDS MySQL db.t2.micro (vCPU:1 e memória de 1 GB de RAM), tipo de implantação Single-AZ, instância sob demanda e armazenamento de uso geral (SSD gp2) com capacidade de 20GB.

Os dados do quadro 4 foram obtidos da Calculadora de Custos da AWS RDS MySQL (AMAZON, 2021e).

**Quadro 4 – Estimativa de custos AWS RDS MySQL**

<b>Estimativa para Amazon® RDS MySQL</b>	<b>Valor (USD)</b>
RDS MySQL: 1 instância x 0,035 USD horas x 730 horas por mês	25,55
Armazenamento de uso geral 20GB (valor mensal)	4,38
RDS MySQL valor (adiantado)	0,00
Total mensal: Armazenamento para cada instância RDS (Uso geral SSD (gp2)), Capacidade (20 GB), Quantidade (1), Tipo instância (db.t2.micro), Tipo implantação (Single-AZ), Instância sob demanda	29,93

Fonte: Elaborado pela autora

Até a data de conclusão desse artigo, não há custos da Amazon® para as transferências de dados da Internet para dentro do AWS RDS, porém foi estimado que não ultrapasse 1GB de transferência de dados do AWS RDS para a Internet, cujo valor é gratuito até 1GB; e os próximos 10 TB/mês, tem-se um custo de 0,15 USD por GB.

O valor previsto para os serviços AWS Lambda e AWS RDS MySQL com 20GB de armazenamento é de 31,33 USD mensal e 375,96 USD anual.

#### 4.3 Análise dos resultados obtidos

O tipo de instância usada para o banco de dados apresentou boa performance com BD vazio, porém com 2GB de dados, já é perceptível um atraso, cerca de 1 minuto para receber o e-mail. Como resposta a alguma análise no BDR, esse tempo tende a aumentar com o aumento do BDR.

A memória alocada na função Lambda e o tempo máximo de processamento usado para cálculo atenderam os requisitos do código, sendo usado 126MB de 128MB disponíveis. Não houve nenhuma taxa de erro durante os testes e os *e-mails* foram recebidos conforme o esperado com a quantidade esperada de alarmes.

O tamanho do BDR alocado de 20GB atende aos requisitos, pois foi calculado que o BDR dados cheio do SCADA é de 2GB, com crescimento de 40 MB por mês. Portanto iria-se ter um *upload* inicial de 2GB e crescimento de aproximadamente 500 MB por ano. É importante avaliar bem os tempos de coleta dos instrumentos para não gerar dados desnecessários, caso contrário, poderia ter-se um crescimento de 4 GB por mês do BDR.

Foi estimado um valor mensal de 31,33 USD, sendo que o processamento do código na função Lambda é de 1,40 USD por mês. Usar o processamento no computador local, em vez da função Lambda na nuvem, não seria uma economia considerável. Além disto o computador local é dedicado ao sistema de automação SCADA. O benefício de ter um *backup* externo no BDR na nuvem protege o sistema caso ocorra algum dano no computador local, além de possuir outras vantagens já mencionadas nesse artigo.

#### 5 CONSIDERAÇÕES FINAIS

Os testes e resultados obtidos foram realizados com um notebook de teste e sinal estável de Internet, os quais foram bastante confiáveis e houve notificação por *e-mails* recebidos quase imediata com o BDR vazio. Porém com 2GB no BDR, já é perceptível um atraso de cerca de 1 minuto, que é o tempo de processamento e análises dos dados no Amazon RDS for MySQL. A conclusão da viabilidade do uso do Amazon RDS for MySQL e da Computação em Nuvem dependerá se o atraso de 1 minuto da informação é considerável ou não. E deve-se levar em conta que o fornecedor da nuvem seja uma empresa confiável, pois caso haja falência, poderá haver perda dos dados.

Numa aplicação real, haverá a necessidade de avaliar a disponibilidade do sinal de celular ou fazer uma redundância com rede *Wi-Fi* na planta industrial, para que o operador não fique numa área em que ocorra o risco de não receber notificações pelo *e-mail*.

O uso de ferramentas de código aberto para desenvolvimento do programa e a implementação em nuvem melhoraram a implantação do projeto, pois não há custos de licenças nem de *hardwares*, havendo apenas o custo do armazenamento, consumo de dados e tempo de processamento apresentados no item 4 deste artigo.

Como sugestão de melhoria, pode utilizar as ferramentas de análise de dados ou visualizações em *dashboards* que a AWS oferece, além de utilizar a função Lambda para implementar códigos para enviar o histórico, ou gráficos dos equipamentos que geraram os alarmes, e dos que possuem correlação ao alarme, para um melhor entendimento da situação do sistema pelo operador por e-mail. Uma próxima abordagem seria testar outro BDR da Amazon® ou usar outra instância do Amazon RDS for MySQL.

A proposta do BDR em nuvem apresentada nesse artigo permitirá o monitoramento em tempo real de vários processos, permitindo futuras integrações entre os sistemas dentro da indústria, além de visibilidade em tempo real da planta *online*, preparando o ambiente para a quarta revolução industrial.

## REFERÊNCIAS

ALCÁCER, V.; CRUZ-MACHADO, V. Scanning the Industry 4.0: a literature review on technologies for manufacturing systems. **Engineering Science and Technology, an International Journal**, v. 22, n. 3, p. 899-919, 2019. Disponível em: <https://www.sciencedirect.com/science/article/pii/S2215098618317750?via%3Dihub>. Acesso em: 1 jul. 2020.

ALVES, Cynthia. **Otimização do sistema de alarmes em uma planta química utilizando conceitos de Lean e Six Sigma**. 2013. Monografia (Graduação em Engenharia Química) – Escola de Engenharia de Lorena, Universidade de São Paulo, Lorena, 2013. Disponível em: <http://sistemas.eel.usp.br/bibliotecas/monografias/2013/MEQ13001.pdf>. Acesso em: 12 jun. 2020.

AMAZON®. **A maior plataforma na nuvem**. 2021a. Disponível em: <https://aws.amazon.com/pt/what-is-aws/>. Acesso em: 17 fev. 2021.

AMAZON®. **MySQL no Amazon RDS**. 2021b. Disponível em: [https://docs.aws.amazon.com/pt\\_br/AmazonRDS/latest/UserGuide/CHAP\\_MySQL.html](https://docs.aws.amazon.com/pt_br/AmazonRDS/latest/UserGuide/CHAP_MySQL.html). Acesso em: 17 fev. 2021.

AMAZON®. **AWS Lambda**. 2021c. Disponível em: <https://aws.amazon.com/lambda>. Acesso em: 17 fev. 2021.

AMAZON®. **AWS Lambda Pricing**. 2021d. Disponível em: [https://aws.amazon.com/lambda/pricing/?nc1=h\\_ls](https://aws.amazon.com/lambda/pricing/?nc1=h_ls). Acesso em: 18 fev. 2021

AMAZON®. **Configure Amazon RDS for MySQL**. 2021e. Disponível em: <https://calculator.aws/#/createCalculator/RDSMySQL>. Acesso em: 18 fev. 2021

CENTRALSERVER. **O que é e como funciona o cloud computing?** 2020. Disponível em: <https://blog.centralserver.com.br/o-que-e-e-como-funciona-o-cloud-computing/>. Acesso em: 1 jul. 2020.

COSTA, Paulo. **Uma abordagem para escolha entre os paradigmas de banco de dados relacional e NoSQL em projetos de software**. Monografia (Graduação em Sistemas de Informação) – Universidade Federal do Rio Grande do Norte, Caicó, RN, 2019. Disponível em: [https://monografias.ufrn.br/jspui/bitstream/123456789/10372/1/UmaAbordagemParaEscolha\\_Costa\\_2019.pdf](https://monografias.ufrn.br/jspui/bitstream/123456789/10372/1/UmaAbordagemParaEscolha_Costa_2019.pdf). Acesso em: 24 fev. 2021

CURINO, C.; JONES, E.; POPA, R.; MALVIYA, N.; WU, E.; MADDEN, S.; BALAKRISHNAN, H.; ZELDOVICH, N. Relational cloud: a database-as-a-service for the cloud. *In: BIENNIAL CONFERENCE ON INNOVATIVE DATA SYSTEMS RESEARCH*, 5, CIDR 2011, January 9-12, 2011 Asilomar, California. 2011. Disponível em: [https://dspace.mit.edu/bitstream/handle/1721.1/62241/Zeldovich\\_Relational%20cloud.pdf?sequence=1&isAllowed=y](https://dspace.mit.edu/bitstream/handle/1721.1/62241/Zeldovich_Relational%20cloud.pdf?sequence=1&isAllowed=y). Acesso em: 22 fev. 2021.

FARIAS, Fred Augusto de Melo. **Avaliação de desempenho entre banco de dados relacionais e NoSQL**. 2014. Monografia (Graduação em Sistemas de Informação) – Universidade Federal da Paraíba, Rio Tinto, PB, 2014. Disponível em: <https://repositorio.ufpb.br/jspui/handle/123456789/17040>. Acesso em: 24 fev. 2021.

FEDAK, Vladimir. **Serverless applications with Lambda: 5 use cases**. 2018. Disponível em: <https://itsvit.com/blog/serverless-applications-with-lambda-5-use-cases/>? Acesso em: 16 fev. 2021.

GRECO, Luca; PIERA, Maresca; CAJA, Jesús. **Big data and advanced analytics in Industry 4.0: a comparative analysis across the European Union**. 2019. Disponível em: <https://www.sciencedirect.com/science/article/pii/S2351978919311102>. Acesso em: 30 mai. 2020.

HAMMES, Dayne; MEDERO, Hiram; MITCHELL, Harrison. Comparison of NoSQL and SQL Databases in the Cloud. 2014. **Proceedings of the Southern Association for Information Systems (SAIS)**, Macon, GA, p. 21-22, 2014. Disponível em: <http://aisel.aisnet.org/sais2014/12>. Acesso em: 23 fev. 2021

HONEYWELL®. **Alarm management standards: are you taking them seriously?** 2011. p.10. Disponível em: <https://www.honeywellprocess.com/library/marketing/whitepapers/honeywell-alarm-manager-alarm-management-standards-taken-seriously-wp817.pdf>. Acesso em: 21 jun. 2020.

IBM®. **O que é computação em nuvem?** 2021. Disponível em: <https://www.ibm.com/br-pt/cloud>. Acesso em: 23 fev. 2021.

INTERNATIONAL SOCIETY OF AUTOMATION (ISA). **ANSI/ISA-101: human machine interfaces for process automation systems**. North Carolina: ISA, 2015.

KOZÁK, Š.; RUŽICKÝ, E.; ŠTEFANOVIČ, J.; SCHINDLER, F. Research and education for industry 4.0: Present development. *In: CYBERNETICS & INFORMATICS (K&I)*. IEEE, 2018, p. 1-8.

LÓSCIO, B. F.; OLIVEIRA, H. R.; PONTES, J. C. NoSQL no desenvolvimento de aplicações web colaborativas. 2011. In: VIII SIMPÓSIO BRASILEIRO DE SISTEMAS COLABORATIVOS (SBSC), 8, v. 10, n. 1, 2011, Paraty, RJ. Disponível em: [https://www.addlabs.uff.br/sbsc\\_site/SBSC2011\\_NoSQL.pdf](https://www.addlabs.uff.br/sbsc_site/SBSC2011_NoSQL.pdf). Acesso em: 24 fev. 2021.

MICROSOFT®. **O que é computação na nuvem?** 2020. Disponível em: <https://azure.microsoft.com/pt-br/overview/what-is-cloud-computing/#cloud-deployment-types>. Acesso em: 21 jun. 2020.

ROCHA, Elmer Henrique. **ANSI/ISA-18.2-2016**: management of alarm systems for the process industries. 2016. Disponível em: [http://isasp.org.br/wp-content/uploads/2020/01/ISA-18\\_2-III-Simp%C3%B3sio-ISA-S%C3%A3o-Paulo-Sabesp-Nov2016.pdf](http://isasp.org.br/wp-content/uploads/2020/01/ISA-18_2-III-Simp%C3%B3sio-ISA-S%C3%A3o-Paulo-Sabesp-Nov2016.pdf). Acesso em: 21 jun. 2020.

SHEHRI, Walled Al. **Cloud database** – database as service. 2013. Disponível em: <http://airccse.org/journal/ijdms/papers/5213ijdms01.pdf>. Acesso em: 22 fev. 2021

ZHANG, Qi; CHENG, Lu; BOUTABA, Raouf. **Cloud computing**: state-of-the-art and research challenges. 2010. Disponível em: <https://link.springer.com/content/pdf/10.1007/s13174-010-0007-6.pdf>. Acesso em: 1 jul. 2020.

## AGRADECIMENTOS

Meus sinceros agradecimentos à minha família que sempre contribuiu para meu constante aprendizado nessa vida.

Aos sobrinhos, por serem minha constante inspiração.

E acima de tudo agradeço a Deus.

Também gostaria de agradecer aos excelentes professores da pós-graduação de Indústria 4.0 da Faculdade SENAI de Tecnologia Mecatrônica. Muito obrigada por tudo, inclusive pela paciência.

E agradeço aos colegas da turma T8 da pós-graduação de Indústria 4.0, que muito enriqueceram as aulas.

Aos meus amigos, agradeço pela amizade e pelo constante apoio.

Meus parabéns à diretoria e coordenação da Faculdade SENAI de Tecnologia e Mecatrônica pela excelente infraestrutura e organização do curso.

O maior agradecimento de todos, à minha mãe, meu exemplo de vida, descanse em paz.

## **SOBRE OS AUTORES**

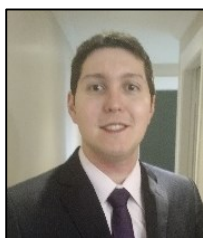
---

### **<sup>i</sup> SYLVIA CRISTINA KIMURA (Aluna)**



Possui graduação em Engenharia de Controle e Automação pela Universidade Federal de Itajubá (2009), pós-graduação em Gestão de Projetos pelo Instituto Mauá de Tecnologia (2015) e atualmente cursando pós-graduação em Indústria 4.0 (2021) pela Faculdade SENAI de Tecnologia e Mecatrônica. Tem experiência na área de automação industrial, em execução de projetos, programação de CP, IHM e atuação em comissionamentos internacionais (EUA, China e Bolívia) e nacionais; e também atuou como supervisora de automação na área de manutenção de automação de sistemas prediais.

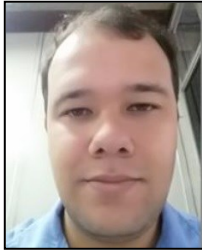
### **<sup>ii</sup> THIAGO TADEU AMICI (Orientador)**



Atualmente ministra aulas na pós-graduação de Indústria 4.0 e na graduação em Tecnologia em Mecatrônica na Faculdade SENAI de Tecnologia Mecatrônica, que fica no SENAI Armando de Arruda Pereira. Assessora também o Instituto SENAI de Tecnologia Metalmeccânica em projetos industriais com foco na Indústria 4.0. Durante 7 anos ministrou aulas pelo SENAI-SP, nos cursos de técnicos de Mecatrônica, Automação Industrial, Eletrônica e Eletroeletrônica, além de Formação Inicial e Continuada (FIC) com cursos voltados ao CP da Siemens. Possui mestrado em Automação e Controle e Processos pelo Instituto Federal de Ciências e Tecnologia de SP (IFSP - 2018), graduação em Engenharia Elétrica pela Faculdade de Engenharia São Paulo (2012), graduação em Tecnologia em Automação Industrial pelo IFSP (2009) e ensino profissionalizante em Eletrônica pela Instituição Liceu de Artes e Ofícios de São Paulo (2002). Tem experiência na área de Engenharia Elétrica, Automação Industrial, Mecatrônica, Robótica e Indústria 4.0. Experiência internacional na aprovação de linha de produção (Cavemil) em Milão na Itália e sua instalação no Brasil. Participou do desenvolvimento do projeto, programação, montagem e apresentação da Linha de Manufatura Avançada Industrial 4.0 realizada em parceria entre o SENAI-SP e a ABIMAQ, que foi exposta na FEIMEC 2018 e da linha de Confecção 4.0, em parceria entre o SENAI-SP e a ABIT.

Endereço para acessar este CV: <http://lattes.cnpq.br/9165856219131658>



**iii JOSÉ ROBERTO DOS SANTOS (Banca)**

Atualmente ministra aulas na pós-graduação de Indústria 4.0 e na graduação em Tecnologia em Mecatrônica na Faculdade SENAI de Tecnologia Mecatrônica, que fica no SENAI Armando de Arruda Pereira. Assessora também o Instituto SENAI de Tecnologia Metalmeccânica em projetos industriais com foco na Indústria 4.0. Durante 9 anos ministrou aulas pelo SENAI-SP, nos cursos de técnico em eletroeletrônica, cursos de aprendizagem industrial eletricitista de manutenção e mecânico de usinagem, além de Formação Inicial e Continuada (FIC) com cursos voltados a área de redes de computadores e programação, possui treinamento de Linux, Cisco e Microsoft. Possui Pós-graduação na área de segurança da informação pela Uninove (2016), graduação em tecnologia da informação e bacharel em sistema da informação (2009), além de superior em Automação industrial. Tem experiência na área de Segurança da informação, administração de ambientes de redes Windows e Linux, e automação industrial.

Endereço para acessar este CV: <http://lattes.cnpq.br/2495692420793433>

**iv DANIEL OTÁVIO TAMBASCO BRUNO (Banca)**

Doutorando em Engenharia da Informação na Universidade Federal do ABC (2016), Mestre em Engenharia da Informação pela Universidade Federal do ABC (2013), Especialista em Banco de Dados pela Universidade de Ribeirão Preto (2007) e Especialista em Educação à Distância pela Universidade Paulista (2012). Possui graduação em Análise de Sistemas pela Universidade Paulista (2003). Revisor do Periódico IEEE Transactions on Medical Imaging. Co-editor, editor de área e revisor da Revista Brasileira de Mecatrônica. É Revisor do periódico ForScience do Instituto Federal de Minas Gerais. Atualmente é Técnico em Manufatura Digital e professor da Escola Senai "Eng. Octávio Marcondes Ferraz" e Professor de ensino superior na Universidade Paulista. Tem experiência na área de Inteligência Artificial, Desenvolvimento de Sistemas de Informação, Processamento de imagens, Implantação e Administração de sistemas Product Lifecycle Management (PLM), Segurança de redes e Internet das Coisas.

Endereço para acessar este CV: <http://lattes.cnpq.br/3491851270517427>