FACULDADE SENAI DE TECNOLOGIA MECATRÔNICA

ANÁLISE DE BANCO DE DADOS ORIENTADO A SÉRIE TEMPORAL NO ARMAZENAMENTO DE DADOS PROVENIENTES DE PROCESSOS INDUSTRIAIS

ANALYSIS OF TIME SERIAL DATABASE IN THE STORAGE OF INDUSTRIAL PROCESSES DATA

João Bosco Pedralino Silva[1], [i]
Daniel Otavio Tambasco Bruno[2], [ii]
Thiago Tadeu Amici[3], [iii]
Paulo Sebastião Ladivez[4],[iv]

RESUMO

O processo de digitalização é a essência da Indústria 4.0. Para tornar este fato possível, surgem em cada vez mais quantidades dispositivos inteligentes, redes de comunicação com protocolos cada vez mais avançados e linguagens de processamento de dados com algoritmos de inteligência artificial. Porém, em meio a este processo, vale ressaltar uma tecnologia de mesma importância e relevância, o banco de dados. Este trabalho consiste em analisar algumas das tecnologias de banco de dados presentes no mercado para atender a uma real necessidade da indústria, que é o armazenamento de dados originados em seus processos. Estes dados possuem características de serem orientados ao tempo, mais conhecidos como dados de série temporal que por sua vez, possuem características onde algumas tecnologias de banco de dados levam vantagem em relações a outras no trato deste tipo de dado. As tecnologias analisadas neste trabalho foram as de banco de dados relacionais, bancos de dados não relacionais e uma extensão especialmente desenvolvida para trabalhar com dados de série temporal. Neste trabalho, foi constatada a importância de se fazer uma análise criteriosa sobre a tecnologia de banco de dado a ser utilizada, de modo que não venha a ser um gargalo ou limitador no sistema de análise de dados.

ABSTRACT

The process of digitalization is the essence of Industry 4.0. To make this fact possible, more and more intelligent devices, communication networks with increasingly advanced protocols, and data processing languages with artificial intelligence algorithms are emerging. However, in the middle of this process, it is worth emphasizing a technology of the same importance and relevance, the database. This work consists of analyzing some of the database technologies present in the market to meet a real need of the industry, which is the storage of data originated in their processes. These data have characteristics of being time oriented, better known as time series data that in turn, have characteristics where some database

[1] Pós-graduado em Indústria 4.0. Pesquisador P&D na empresa Nadir Figueiredo. E-mail: joaoboscops@gmail.com
[2] Mestre em Engenharia da Informação. Professor da Faculdade SENAI de Tecnologia Mecatrônica Industria. E-mail: daniel.bruno@sp.senai.br
[3]Mestre em Controle e Automação de Processos. Professor da Faculdade SENAI de Tecnologia Mecatrônica. E-mail: thiago.amici@sp.senai.br
[4]Engenheiro e Professor Especialista da Faculdade SENAI de Tecnologia Mecatrônica. E-mail: paulo.ladivez@sp.senai.br

technologies take advantage in relations to others in the treatment of this type of data. The technologies analyzed in this work were relational database, nonrelational databases and a specially developed extension to work with time series data. In this job, it was noted that it is important to make a careful analysis of the database technology to be used so that it does not become a bottleneck or limiter in the data analysis system.

Data de submissão: (16/05/2019)
Data de aprovação: (20/06/2019)

1 INTRODUÇÃO

A indústria 4.0 nos remete ao conceito da digitalização de quase tudo. De acordo com Brynjolfsson e McAfee (2015), a digitalização de quase tudo é um dos fenômenos mais importantes dos últimos anos. Através dos sistemas ciber-físicos em constante expansão, é notório que a quantidade de dados de processos industriais, com potencial de serem digitalizados, alcançaram um patamar muito alto atualmente. Os dados têm ganhado tanta importância ultimamente, que segundo a entrevista realizada com o então presidente da Intel no Brasil, Maurício Ruíz, por Loureiro (2018) recebeu o título de “Os dados são o novo petróleo”.

É possível acompanhar em literaturas especializadas a vasta disponibilidade de sensores, atuadores, motores e dispositivos voltados à automação industrial, que já se encontram preparados para o cenário da Indústria 4.0, ou seja, estão preparados para disponibilizar dados de processos e metadados do próprio equipamento, com a finalidade de serem utilizados em outras camadas da automação industrial alinhadas aos conceitos da Indústria 4.0.

Na outra ponta do processo de dados, também podemos acompanhar a crescente utilização de técnicas de análises de dados, como *data mining, machine learning, deep learning,* além de outras tecnologias que compõe a inteligência artificial (IA), todas com a finalidade de extrair real valor dos dados, de forma que agreguem valor aos processos e negócios.

No meio deste processo de disponibilização e processamento de dados, se faz necessário a utilização de tecnologias de armazenamento e recuperação de dados, entrando em cena os sistemas de gerenciamento de banco de dados (SGBD). Os SGBDs mais populares são os tradicionais bancos de dados relacionais que utilizam a linguagem *Structed Query Language* (SQL). O problema é que esta tecnologia de banco de dados, não apresenta ferramentas e recursos satisfatórios para o trato de dados temporais. Segundo Cassol (2012, p. 13), este afirma que “Entretanto, bancos de dados tradicionais não tem um suporte amplo para armazenamento e consulta sobre este tipo de dados eficientemente [...]”.

Também será realizada a comparação entre algumas tecnologias de banco de dados presentes no mercado, como banco de dados relacionais e não relacionais e banco de dados orientado a séries temporais de dados, demonstrando as principais vantagens e desvantagens de cada uma delas, bem como a diferenças entre elas, utilizando como objeto de estudo dados de série temporal proveniente de processos industriais.

Nota-se que são poucas as literaturas que focam no armazenamento de dados para esta finalidade. Para Cassol (2012, p. 13), “O problema é que não há estudo que analise as escolhas de *design* disponíveis, fazendo uma análise através de dados concretos”.

Tão importante quanto definir as topologias de redes e protocolos para aquisição de dados, e as linguagens de programação e algoritmos a ser utilizados na análise dos mesmos, também é importante a definição de tecnologias de armazenamento e recuperação destes dados.

O objetivo deste trabalho é apresentar por meio de revisão de literatura, os principais conceitos de banco de dados relacionais, popularmente conhecidos como bancos de dados SQL e os bancos não relacionais, conhecidos como bancos de dados *Not Only SQL* (NoSQL).

Este trabalho espera auxiliar na seleção da tecnologia de banco de dados mais adequada a ser implementada no armazenamento de dados provenientes de processos industriais, respeitando a natureza desses dados e considerando a utilização destes em uma análise posterior.

Também será abordado em mais detalhes bancos de dados orientados a série de dados temporal, dados estes que correspondem a natureza de grande parte dos dados oriundos dos processos industriais.

Por fim, serão apresentadas as vantagens e desvantagens de se utilizar bancos de dados SQL e NoSQL, quando comparados entre si e as vantagens de se utilizar um banco de dados orientado a série temporal para armazenamento de dados originados em processos industriais.

2 FUNDAMENTAÇÃO TEÓRICA

A seguir serão apresentadas algumas das tecnologias presentes no conceito da Indústria 4.0.

2.1 **Internet of Things (IoT)**

Ao analisarmos o processo de gerenciamento de dados industriais, o *Internet of Things* (IoT) ou sua derivação *Industrial Internet of Things* (IIoT) estariam localizadas na ponta de entrada do processo de digitalização, já que sua rede proporciona a leitura de dados originados em dispositivos industriais.

Não há uma única definição disponível para IoT aceita pela comunidade mundial de usuários, já que existem diferentes grupos incluindo acadêmicos, pesquisadores, práticos, inovadores, desenvolvedores e a própria corporação que definiu o termo. (MADAKAM; RAMASWAMY; TRIPATHI, 2015).

Para Madakam; Ramaswamy e Tripathi (2015, p. 165), a melhor definição de IoT pode ser “Uma rede aberta e abrangente de objetos inteligentes que têm a capacidade de se auto organizar, compartilhar informações, dados e recursos, reagindo e agindo diante de situações e mudanças no ambiente”. (tradução nossa).[5]

Ainda segundo Madakam; Ramaswamy e Tripathi (2015, p. 165) o IoT pode ser considerado uma rede global que permite a comunicação entre humanos e humanos, humanos e coisas, coisas e coisas, oferecendo uma identidade única para cada ator nesta rede. A ideia do estar conectado vai além de utilizarmos servidores, computadores, *tablets* ou *smartphones.* No conceito de IoT, sensores e atuadores embarcados nos mais diversos equipamentos e dispositivos estão interligados através de meios com ou sem fio, utilizando

[5] “An open and comprehensive network of intelligent objects that have the capacity to auto-organize, share information, data and resources, reacting and acting in face of situations and changes in the environment”.

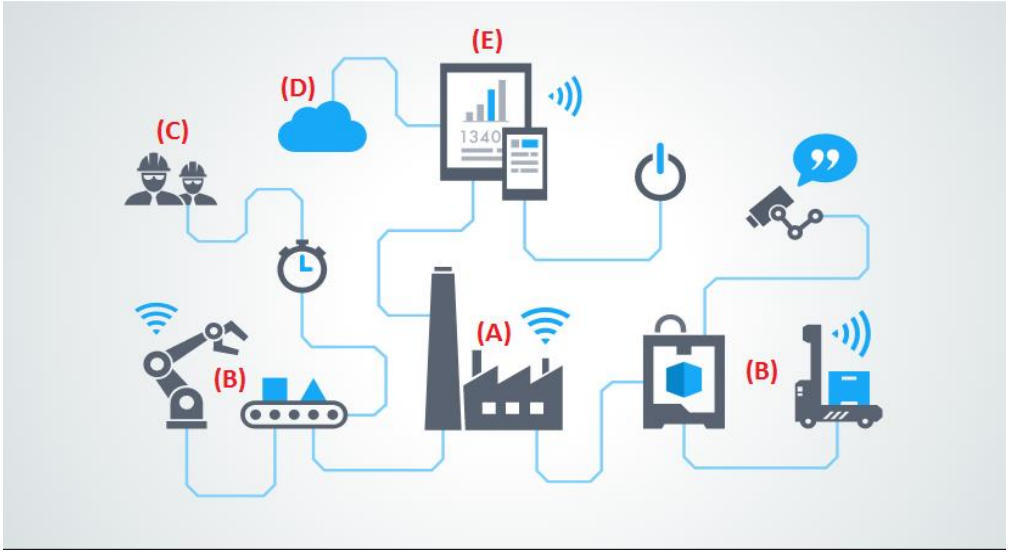
os mesmos endereços de *Internet Protocol* (IP) utilizados na Internet. Estas redes despejam um grande volume de dados que seguem para computadores para serem analisados.

Dentro do contexto da Indústria 4.0, se faz importante ressaltar qual a diferença em IoT e IIoT. IoT é uma rede baseada em Internet que conecta objetos físicos (coisas) que podem se comunicar umas com as outras e com outros sistemas. Por sua vez, IIoT faz referência ao IoT, porém dentro de um contexto industrial. Este contexto se refere a conectar máquinas e gerenciar dados para permitir melhorias na produtividade e qualidade. Para finalizar, IoT tende a ser centrada no consumidor, já IIoT foca na melhoria da eficiência da produção, da cadeia de suprimentos e contextos de gerenciamento. (TULIP, 2018)

A figura 1 é uma representação do IIoT, onde pode-se notar uma indústria qualquer identificada pela letra A. No mesmo nível da indústria, identificado pela letra B, é possível perceber a comunicação entre robôs e transportadores, e a comunicação entre dispositivos e equipamentos de armazenamento e expedição. A letra C representa a parte humana conectada ao sistema, podendo ser interpretada como operadores, engenheiros e analistas de dados. A representação de um cronometro conectando pessoas, máquinas e equipamentos do processo, significa que tal comunicação e troca de dados é feito em tempo real. Já letra D representa a computação ou comunicação em nuvem. Isto porque todo ou parte destes dados podem ser processados e, ou armazenados em sistema baseados em computação em nuvem, bem como serem utilizados por outros sistemas em uma localização remota à indústria representada na imagem. Por fim, a letra E representa a interface de exibição dos dados processados e, ou armazenados na nuvem, permitindo também que alguns controles do processo produtivo possam ser executados de forma remota.

Como se pode observar, a figura 1 faz uma simples representação do IIoT, onde se pode observar a comunicação entre máquinas (coisas), entre máquinas e humanos, e entre sistemas.

Figura 1 - Representação do IIoT



Fonte: Tulip (2018)

2.2 Análise de dados (Big Data e Data Analytics)

Para Coelho (2016), dentre as principais questões levantadas pelas grandes empresas tecnológicas está o como processar dados para que tenham significado, permitindo que as organizações melhorem as operações com decisões mais rápidas e inteligentes.

Esta questão faz referência direta ao conceito de *Big Data* e *Data Analytics*, que segundo Coelho:

> "[...] refere-se a grandes quantidade de dados que são armazenados a cada instante resultante da existência de milhões de sistemas atualmente ligados à rede (IoT), que produzem dados em tempo real sobre quase tudo [...]" (COELHO, 2016, p. 22).

Pode-se perceber uma relação direta entre os conceitos de IoT e o do *Big Data* e *Data Analytics*, onde o primeiro é responsável por gerar dados e o segundo e o terceiro por trabalhar estes dados, armazenando e processando. Porém este processo de dados deve ser realizado com métricas bem estabelecidas, de forma que resulte em uma análise destes dados. Ainda de acordo com Coelho (2016, p. 23), "Com tantos dados a serem gerados continuamente são precisas ferramentas de análise poderosas para lhe dar significado". E segue "Podem ser verificados e validados, contudo não tem qualquer significado se não forem interpretados e contextualizados, dando origem a informação."

Após perceber a importância do IoT na geração de dados e do *Big Data* e *Data Analytics* na análise destes dados, outro elo central de grande importância nesta cadeia e que faz parte do conceito de *Big Data* é o armazenamento de dados. Pois outra questão importante para a empresas tecnológicas é onde guardar os dados de forma segura e acessá-los de qualquer lugar (COELHO, 2016).

2.3 **Computação em nuvem**

A computação em nuvem é utilizada para armazenar e analisar uma grande quantidade de dados típicas de aplicações da Indústria 4.0. No conceito de computação em nuvem, os recursos são fornecidos através de utilitários ou ferramentas cobradas por demanda e liberados aos usuários através da Internet. Os recursos da computação em nuvem são compartilhados particionando recursos físicos através de tecnologias de virtualização (BORTOLINI; et. al., 2017, p. 5703).

A figura 2 possui certa similaridade com a figura 1, mas o enfoque da figura 2 se diferencia por destacar o funcionamento de serviços em nuvem. A letra A representa o elo central desta cadeia, representando os serviços em nuvem propriamente ditos. Esta figura tem o objetivo de mostrar as vantagens dos serviços de nuvem aliados ao IoT. Pode-se notar na letra B uma etapa mais rural de uma determinada cadeia produtiva com dados acessados em tempo real. Já na letra C percebe-se que uma fábrica compartilha dados com o setor rural, através de serviços em nuvem. Por sua vez, a fábrica possui parte de seus ativos interconectados, alimentando a nuvem com dados de produção. A letra D representa a parte logística desta cadeia e a interação com outros elos desta cadeia.

De forma resumida, através dos serviços de nuvem aliado à IoT, é possível conectar toda a cadeia produtiva, desde a produção rural até o beneficiamento destes produtos através de processos industriais, alinhados aos objetivos da Indústria 4.0 que é a melhora da gestão, eficiência produtiva e cadeia de suprimentos.

Figura 2 - Representação de computação em nuvem

Fonte: Portal Lubes (2018).

Existem várias empresas no mercado que fornecem serviços de computação em nuvem. Dentre elas de destacam-se a Amazon® com o seu *Amazon Web Services*® (AWS), a Microsoft® com o Azure®, a IBM® com o Watson®. Também nota-se este tipo de serviço sendo fornecido por empresas que possuem forte atuação e tradição na área industrial, como a Siemens® através do seu MindSphere®.

2.4 **Manipulação de dados**

As tecnologias de banco de dados são parte fundamental no sistema de aquisição e análise de dados. Como já citado na introdução deste trabalho, a entrevista feita com o então presidente da Intel® no Brasil, Maurício Ruíz, os dados vêm se tornando cada vez mais relevantes e já vêm sendo chamado de o novo petróleo. Assim como o petróleo, os dados precisam ser tratados e transformados para agregarem valor a um processo ou negócio. Esta transformação nos dados, consiste em obter informação e conhecimento, e antes de serem tratados, a forma como são armazenados terá impacto direto na qualidade da informação gerada.

Atualmente, podemos notar em várias soluções de armazenamento, a utilização do já consolidado banco de dados relacional, bem como a crescente utilização de bancos de dados não relacionais ou especializados. Para Teorey et al.:

> Embora muitos sistemas de bancos de dados especializados (orientados a objeto, espaciais, multimídia etc.) tenham encontrado substanciais comunidade de usuários nos campos da ciência e engenharia, os sistemas relacionais continuam sendo a tecnologia de banco de dados dominantes nas empresas comerciais (TEOREY et al., 2014, I. 315).

A figura 3, ilustra um *ranking* com os dez bancos de dados mais utilizados no momento:

Figura 3 – Ranking de utilização dos bancos de dados

347 systems in ranking, May 2019

Rank			DBMS	Database Model	Score		
May 2019	Apr 2019	May 2018			May 2019	Apr 2019	May 2018
1.	1.	1.	Oracle	Relational, Multi-model	1285.55	+5.61	-4.87
2.	2.	2.	MySQL	Relational, Multi-model	1218.96	+3.82	-4.38
3.	3.	3.	Microsoft SQL Server	Relational, Multi-model	1072.19	+12.23	-13.66
4.	4.	4.	PostgreSQL	Relational, Multi-model	478.89	+0.17	+77.99
5.	5.	5.	MongoDB	Document	408.07	+6.10	+65.96
6.	6.	6.	IBM Db2	Relational, Multi-model	174.44	-1.61	-11.17
7.	↑ 8.	↑ 9.	Elasticsearch	Search engine, Multi-model	148.62	+2.62	+18.18
8.	↓ 7.	↓ 7.	Redis	Key-value, Multi-model	148.40	+2.03	+13.06
9.	9.	↓ 8.	Microsoft Access	Relational	143.78	-0.87	+10.67
10.	↑ 11.	10.	Cassandra	Wide column	125.72	+2.11	+7.89

Fonte: DB-Engines (2019)

Nesta figura, observa-se que entre os 347 sistemas de banco de dados ranqueados, entre os 10 mais utilizados, seis são do tipo relacional (Oracle®, MySQL®, Microsoft SQL Server®, PostgreSQL®, IBM Db2®, Microsoft Access®). Os quatro tipos não relacionais são MongoDB®, Elasticsearch®, Redis® e Cassandra®. A figura 3 também exibe a tendência de crescimento na utilização destes bancos de dados, tendo sua posição alterada no *ranking* os bancos Elasticsearch® e Cassandra®; bem como uma tendência de queda dos bancos Redis® e Microsoft Access®, utilizando como referência os meses de maio e abril de 2019 e maio de 2018.

2.4.1 ***Dados temporais***

Segundo Silberschatz; Korth e Sudarshan (2012), banco de dados modelam aspectos do mundo real. Normalmente estes bancos modelam apenas um estado do mundo real, o estado atual. Quando o estado sobre determinado aspecto é atualizado, os dados do estado anterior são perdidos. Porém em muitas aplicações se faz necessário armazenar e recuperar dados do antigo estado, como aplicações industriais, por exemplo:

> Porém, em muitas aplicações, é importante armazenar e recuperar informações sobre estados passados. [...] Um sistema de monitoração de fábrica pode armazenar informações sobre as leituras atual e passada dos sensores de fábrica, para análise. (SILBERSCHATZ; KORTH; SUDARSHAN 2012, p. 670).

Ainda de acordo com Silberschatz; Korth e Sudarshan (2012), estes tipos de bancos de dados são chamados de bancos de dados temporais.

Para Cassol (2012), “Dados temporais podem ser fundamentalmente de três tipos: instantes, intervalos ou períodos” (CASSOL, 2012, p. 18, apud SNODGRASS, 1999-a). Segue em sua explicação, dizendo que dados de instantes são aqueles que possuem um ponto específico no tempo, como por exemplo, 14/05/2019. Já os dados de intervalos são aqueles de proporções não específicas e que possuem direção cronológica, ou seja, podem ser referentes ao passado ou ao futuro. Um intervalo de tempo poderia ser descrito, por exemplo, como

daqui a seis meses ou três meses atrás. Por fim, períodos são porções de tempo que possuem início e fim bem definidos, iniciando em um instante e terminando em outro no futuro.

Conforme Greenfield (2012), dados de processos, como a temperatura medida por uma sonda, normalmente são armazenados em um banco de dados histórico ou de série temporal e que a conexão entre os dados em um banco de dados de série temporal é o tempo.

2.4.2 ***Dados relacionais***

Dados de produção como ordens de produção, receitas, produtos etc., são tipicamente armazenados em um banco de dados relacional (GREENFIELD, 2012). Em um banco de dados relacional, os relacionamentos representam associações entre uma ou mais entidades (tabelas) de um banco de dados, que é muito utilizado em um sistema de dados que envolvem operações de natureza transacional.

2.4.3 ***Dados temporais versus relacionais***

Para Greenfield (2012), banco de dados historiador tipo série temporal, não trabalham muito bem com dados relacionais da mesma forma que bancos de dados relacionais não trabalham muito bem com uma grande quantidade de dados não estruturados do tipo série temporal. Conclui Greenfield (2012), que por conta de suas diferenças de propósito, ambas tecnologias têm lugar na infraestrutura de Tecnologia da Informação (TI) industrial.

Desta forma, percebe-se que para um cenário de sistema de dados industriais, as tecnologias para dados temporais quanto as tecnologias para dados relacionais, não são concorrentes entre si, podendo ser aliadas e complementares em um projeto de banco de dados industrial mais abrangente.

2.4.4 ***Banco de dados relacionais (SQL) e não relacionais (NoSQL)***

De acordo com Silberschatz; Korth e Sudarshan (2012)

> "Um sistema de banco de dados (SGBD) é uma coleção de dados inter-relacionados e um conjunto de programas para acessar esses dados. A coleção de dados, normalmente conhecida como banco de dados, contém informações relevantes para uma empresa. O principal objetivo de um SGBD é proporcionar uma forma de armazenar e recuperar informações de um banco de dados de maneira conveniente e eficiente." (SILBERSCHATZ; KORTH; SUDARSHAN 2012, p.1).

Após este definição, veremos a seguir alguns tipos de banco de dados.

2.4.5 ***Bancos de dados não relacionais (NoSQL)***

Bancos de dados não relacionais, também são conhecidos como bancos NoSQL. Segundo Lócio et al. (2012), o banco de dados NoSQL "[...] foi uma proposta com o objetivo de atender aos requisitos de gerenciamento de grandes volumes de dados, semiestruturados ou não estruturados que necessitam de alta disponibilidade e escalabilidade" (LÓCIO et al., 2012, p. 1). Uma das principais características dos bancos de dados NoSQL, ainda segundo Lócio et al. (2012), é o fato de ele possuir ausência de esquema ou esquema flexível, que é

ausência completa ou quase total do esquema que define a estrutura dos dados modelados. Isso significa que dados de uma mesma coleção podem ter atributos diferentes.

A seguir as principais características de dois tipos de banco de dados NoSQL que constam no *ranking* no DB-Engines, o MongoDB® e o Cassandra®, conforme demonstrado da figura 3.

2.4.5.1 MongoDB®

MongoDB é um banco de dados orientado a documentos, não usa esquemas relacionais armazenando documentos no formato *Java Script Object Notation* (JSON) (MATOS, 2018).

Segundo MongoDB® (2019), os campos em um documento armazenado podem variar de documento para documento, além de permitir que a estrutura de um documento possa ser alterada ao longo do tempo sem a necessidade de alterar a estrutura de outros documentos.

A seguir exemplo de comandos para inserção de dados em um banco MongoDB:

Quadro 1 - Exemplo de inserção de dados no MongoDB

Exemplo de comandos para inserção de dados no MongoDB	var insertDocuments = function(db, callback){
	return co(function*(){
	const results = yield db.collection('restaurants').insertMany([
	
	{
	"name": "Sun Bakery Tratoria",
	"stars": 4,
	"categories": [
	"Pizza", "Pasta", "Italian", "Coffe", "Sandwiches"
	]
	},{
	"name": "Blue Bagels Grill",
	"stars": 3,
	"categories": [
	"Bagels", "Cookies", "Sandwiches"
	]
	},{
	"name": "Hot Bakery Cafe",
	"stars": 4,
	"categories": [
	"Bakery", "Cafe", "Coffee", "Dessert"
	]
	}

Fonte: MongoDB (2019)

2.4.5.2 Cassandra®

Cassandra® é um banco de dados originalmente desenvolvido pelo Facebook®. Possui um mecanismo de dados descentralizado, distribuído e orientado a coluna (MATOS, 2018).

Uma característica interessante do Cassandra®, quando comparado a um banco SQL convencional, é o fato de que um registro não é composto por várias colunas, pois as colunas ficam internas as linhas da tabela. Isto demonstra a flexibilidade deste banco de dados, pois em uma mesma tabela, podemos possuir dados com tamanho de colunas diferentes.

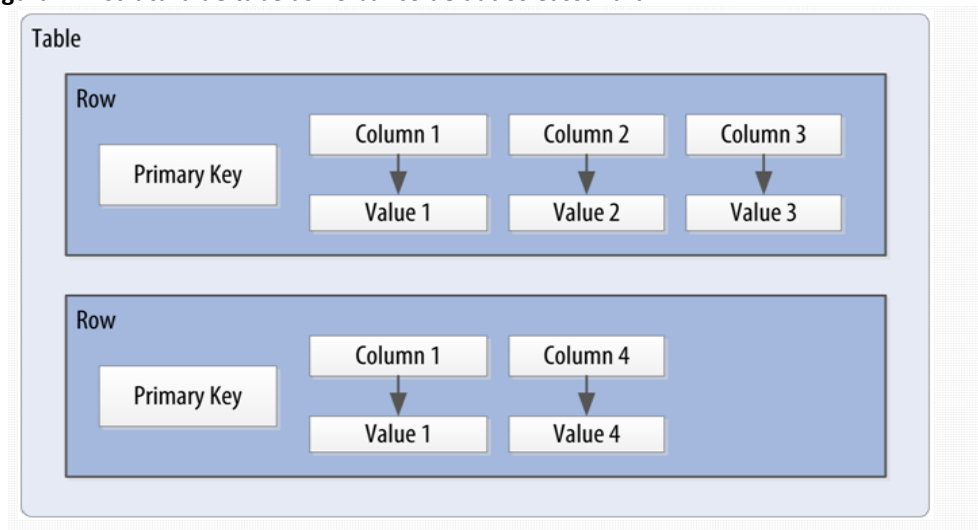
De acordo com Queiros (2017):

> Agora, diferente de uma tabela de um banco relacional, no Cassandra não é necessário popular todas as colunas ao criar uma nova linha, de forma que podemos ter "larguras" diferentes para cada uma delas (uma coluna de um banco de dados

> relacional sempre precisa ser preenchida com algum valor, mesmo que seja um valor nulo. (QUEIROS, 2017)

A figura 4 ilustra esta estrutura.

Figura 4 - Estrutura de tabelas no banco de dados Cassandra



Fonte: Queiros (2017)

Outra característica relevante do Cassandra®, é que qualquer valor registrado em uma coluna possui um valor de estampa do tempo associado, indicando o horário exato em que o registro foi efetuado (QUEIROS, 2017). Esta característica se torna importante porque dados de estampa do tempo são fundamentais em estrutura de bancos de dados que trabalham com dados de séries temporal, como o histórico de um processo industrial, por exemplo.

O Cassandra® possui uma linguagem própria chamada Cassandra *Query Language* (CQL).

2.5 Bancos de dados relacionais (SQL)

Para Silberschatz; Korth e Sudarshan (2012, p. 23), "O modelo relacional [...] usa um conjunto de tabelas para representar tanto os dados quanto a relação entre eles".

Como já citado anteriormente, bancos de dados relacionais não possuem nativamente ferramentas que trabalhem bem com dados de série temporal. Também, em comparação com bancos de dados NoSQL, não possuem esquemas flexíveis, o que pode ser um problema se considerarmos a grande quantidade de estruturas de dados (ou falta delas) passíveis de armazenamento atualmente.

Porém uma alternativa para estes problemas podem ser algumas extensões criadas para mitigar ou eliminar esses problemas, que se apresentam devido à natureza funcional dos bancos de dados relacionais.

A seguir, será apresentada uma extensão chamada TimescaleDB®, desenvolvida para trabalhar sobre o banco de dados relacional PostgreSQL®.

2.5.1 PostgreSQL

Conforme Silberschatz; Korth e Sudarshan (2012, p. 711), “O PostgreSQL é um sistema de gerenciamento de banco de dados objeto-relacional de código fonte aberto”.

Ainda de acordo com Silberschatz; Korth e Sudarshan (2012), o PostgreSQL® possui as principais características de um banco de dados relacional, como consultas complexas, chaves estrangeiras, *triggers, views,* integridades transacionais, pesquisa de texto completo e replicação de dados limitada. Além disso, permite que usuários possam estender o PostgreSQL® com novos tipos de dados, funções, operadores ou método de índices. Finaliza com a informação de que a licença do PostgreSQL® é o do tipo Berkeley *Software Distribution* (BSD), que permite qualquer pessoa usar, modificar, e distribuir o código e documentação para qualquer finalidade gratuitamente.

Estas condições podem ter sido decisivas na escolha do PostgreSQL® como base da extensão TimescaleDB®.

3 BANCO DE DADOS TIMESCALEDB

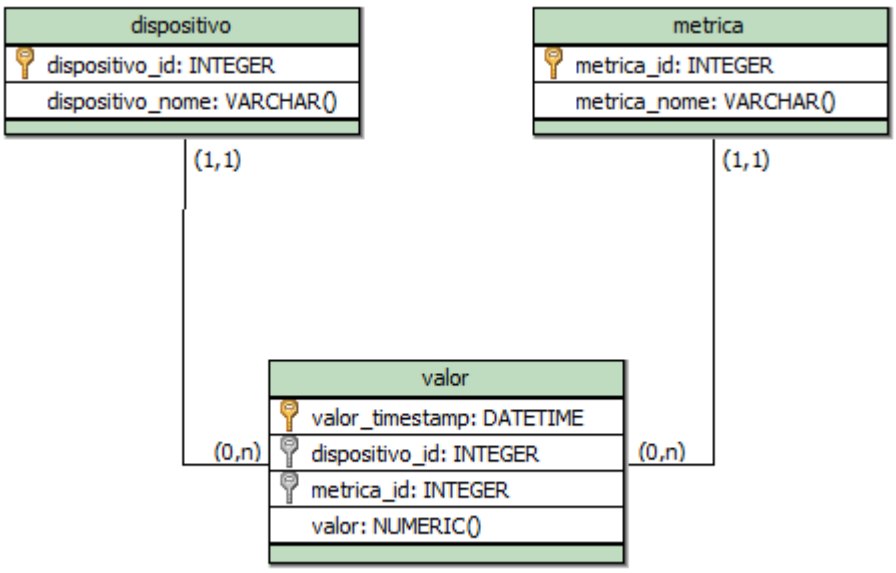
Segundo informações do site oficial, o TimescaleDB® é um banco de dados de série temporal de código aberto, otimizado para rápida ingestão de dados e consultas complexas. Quando comparado com bancos de dados relacionais e NoSQL, o TimescaleDB® oferece o melhor dos dois mundos para os dados de série temporal. (TIMESCALE, 2019).

3.1 Modelo de dados do TimescaleDB®

Banco de dados de série temporal, tipicamente adotam o conceito de *narrow-table* no modelamento de seus dados (TIMESCALE, 2019).

A figura 5 exibe um exemplo simples de modelamento de dados utilizando o conceito *narrow-table.*

Figura 5 - Modelagem de dados utilizando conceito *narrow-table*



Fonte: Autor

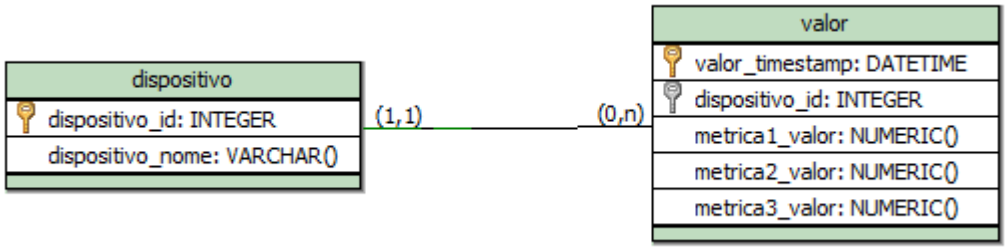
Para melhor exemplificar a imagem acima, tem-se um sensor de temperatura de algum processo industrial qualquer. A tabela responsável por armazenar as métricas do sensor, podem armazenar métricas de diferentes tipos, como por exemplo um sensor de temperatura que pode disponibilizar métricas do próprio dispositivo (quantidade de vezes que o sensor foi acionado) e métricas do processo (valor da temperatura medida). Neste exemplo, para cada um dos dois registros armazenados na tabela "valor" (um para a quantidade de acionamentos e outro para o valor da temperatura), uma estampa do tempo diferente foi relacionada para cada um destes registros.

Este modelo oferece mais flexibilidade na inclusão de métricas, pois ao se registrar dados de um determinado dispositivo na tabela "valor", só serão relacionadas as métricas pertinentes a este sensor e as demais podem ser ignoradas, economizando espaço no banco de dados. Em contrapartida, oferece certa dificuldade quando se faz necessário relacionar registros deste dispositivo a uma mesma estampa do tempo.

Este conceito de armazenamento só faz sentido se os dados forem analisados separadamente. Agora, se o que se deseja saber é qual era o valor da temperatura e da quantidade de acionamentos em um instante específico (em um mesmo *timestamp),* este modelo não é o mais adequado. Além disso, consultas que envolvem muitas métricas diferentes tendem a se tornar mais complexas neste modelo. (TIMESCALE, 2019).

Uma alternativa a este cenário, é a utilização do modelo *wide-table,* conforme pode ser observado na figura 5.

Figura 5 - Modelagem de dados utilizando o conceito *wide-table*



Fonte: Autor.

Diferentemente do modelo anterior, neste modelo as tabelas de métricas e valores são unificadas em uma única tabela, onde as métricas passam a ser colunas da tabela "valor".

Neste caso, temos uma única estampa do tempo para todas as métricas de um determinado sensor, pois todas as métricas são armazenadas em um mesmo registro, garantindo de forma intrínseca a relação entre os dados, sem a necessidade de execução de consultas complexas. Porém, se perde a flexibilidade oferecida no modelo *narrow-table.* Isto porque para cada métrica inserida como atributo na tabela "valor", um valor deve ser inserido nesta métrica, mesmo que seja nulo, ocupando espaço desnecessário de armazenamento no banco de dado.

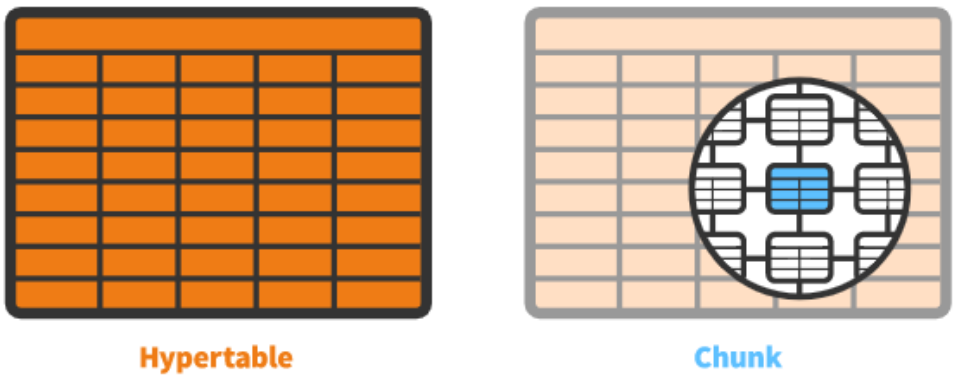
Porém não se trata de um modelo novo e é o que normalmente já se encontra em bancos de dados relacionais quando se faz necessário armazenar dados de série temporal. De qualquer forma, o TimescaleDB® está preparado para trabalhar com os dois tipos de modelos. (TIMESCALE, 2019).

3.2 Arquitetura

De forma geral, do ponto de vista do usuário, o TimescaleDB® expõe o que parece ser tabelas singulares, chamadas *hypertables,* que na verdade, tratam-se de uma abstração de muitas outras tabelas, chamadas *chunks.* Todas as interações do usuário com os dados são feitas através das *hypertables.* (TIMESCALE, 2019).

Os *chunks* são criados particionando os dados da *hypertable* em uma ou várias dimensões. Todos os *hypertables* são particionados por um intervalo de tempo, além de poder ser particionado através de outros atributos, como o código de identificação de uma dispositivo, por exemplo. Por vezes, esse particionamento é referido como particionamento através do espaço-tempo. (TIMESCALE, 2019).

Figura 6 - *Hypertable* e *chunks*



Fonte: Timescale (2019).

Internamente, o TimescaleDB® divide automaticamente as *hypertables* em *chunks,* onde cada *chunk* corresponde a um intervalo de tempo específico e uma região do espaço de partição utilizando *hashing.* (TIMESCALE, 2019).

Segundo Silberschatz; Korth e Sudarshan (2012):

> O *hashing* pode ser usado para duas finalidades diferentes. Em uma organização de arquivo por *hash*, obtemos o endereço do bloco de disco contendo um registro desejado diretamente calculando uma função sobre o valor da chave de busca do registro. Em uma outra organização de índice por *hash,* organizamos as chaves de busca, com seus ponteiros associados, para uma estrutura de arquivo de *hash.* (SILBERSCHATZ; KORTH; SUDARSHAN 2012, p. 319).

Cada *chunk* é implementado utilizando uma tabela padrão de banco de dados, pois internamente no PostgreSQL®, cada *chunk* é uma tabela filho de uma tabela pai *hypertable.* Além disso, os *chunks* são dimensionados corretamente, de modo que todas as árvores B dos índices de uma tabela possam residir na memória durante as inserções. Isso evita problemas ao modificar locais nesta árvore de forma arbitrária. (TIMESCALE, 2019).

Para Silberschatz; Korth e Sudarshan (2012), a estrutura de índice de árvore B apresenta uma alternativa ao problema de diminuição de desempenho ocasionado pelos arquivos de índice organizados de forma sequencial. Esse problema se apresenta quando estes arquivos crescem muito, prejudicando tanto as pesquisas no índice quanto a varredura sequencial de dados. Já a estrutura de árvore B é a mais utilizada das várias estruturas de índice, já que consegue manter sua eficiência apesar das inserções e exclusões de dados.

O TimescaleDB® foi concebido para executar o particionamento de *hypertables* tanto em sistemas *single node* quanto em *clustering,* porém a versão para esta última continua em desenvolvimento. Embora o sistema de particionamento tradicionalmente seja utilizado apenas em sistemas em *clustering,* o TimescaleDB® permite altas taxas de gravação em sistemas *single node.* A versão *single node* do TimescaleDB® foi testada com mais de 10 milhões de linhas em uma *hypertable* sem perda de desempenho (TIMESCALE, 2019).

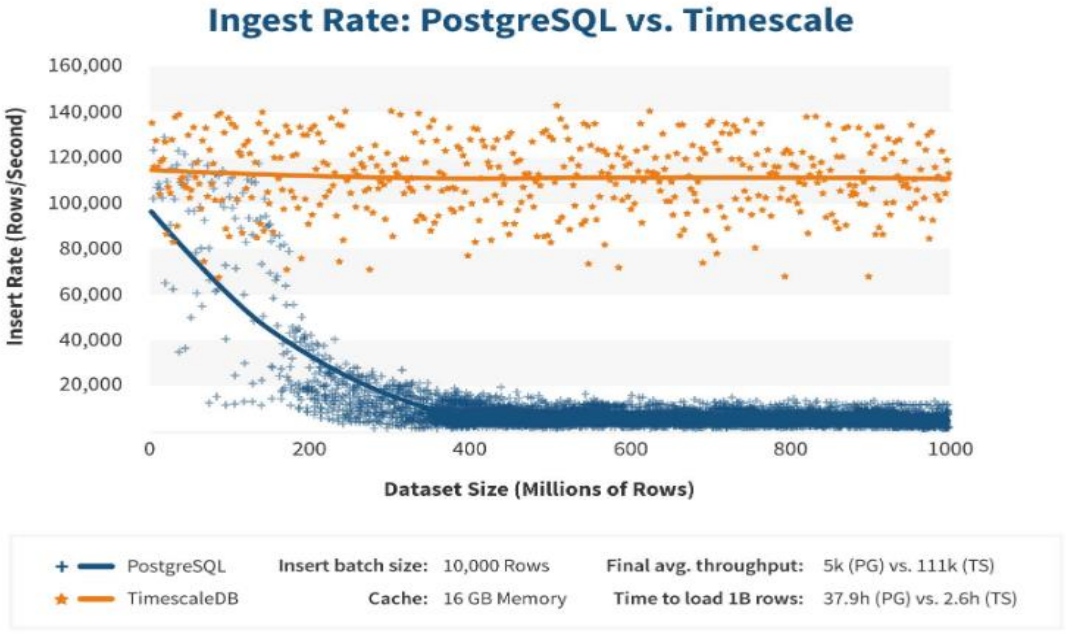
O conceito de *chunks* e *hypertable* apresentado pelo TimescaleDB®, surge como uma alternativa aos modelos de dados baseados em *narrow-table* e *wide-table*, oferecendo o melhor dos dois conceitos, a flexibilidade do *narrow-table* e a indexação ao tempo do *wide-table.*

3.3 TimescaleDB® sobre o PostgreSQL®

A taxa de ingestão de dados é muito maior e mais estável quando se utiliza a extensão TimescaleDB® sobre o PostgreSQL® do que quando se utiliza o PostgreSQL® sem a extensão. Isso ocorre principalmente quando tabelas indexadas do PostgreSQL® não cabem mais na memória. O TimescaleDB® lida com este problema fazendo uma intensa utilização do particionamento espaço-tempo, mesmo quando utilizado em uma única máquina. (TIMESCALE, 2019).

A figura 7 demonstra uma comparação entre um banco de dados PostgreSQL® utilizando a extensão TimescaleDB® com um banco PostgreSQL® sem a extensão. A comparação emula um cenário de monitoramento comum, com clientes de banco de dados inserindo dados em lotes de tamanho moderado orientados ao tempo, com um conjunto de dados de um dispositivo utilizando dez métricas, em armazenamento de 1 bilhão de linhas (em uma única máquina). A experiência foi realizada utilizando uma máquina virtual padrão do Microsoft Azure® (DS4 v2, 8 core) com armazenamento de *Solid-State Drive* (SSD) conectado à rede. (TIMESCALE, 2019).

Figura 7 – Taxa de ingestão de dados: PostgreSQL vs. Timescale



Fonte: Timescale (2019).

Podemos observar que tanto o PostgreSQL® quanto TimescaleDB® começam com a mesma taxa de transferência, 106K e 114K respectivamente para as primeiras solicitações. No entanto, com cerca de 50 milhões de linhas, o desempenho do PostgreSQL® começa a cair rapidamente. Sua média nas últimas 100 mil linhas é de apenas 5 mil linhas/segundo, enquanto o TimescaleDB mantém sua taxa de transferência de 111 mil linhas/segundo. (TIMESCALE, 2019).

3.4 Comparando TimescaleDB® com bancos de dados NoSQL

Quando comparado com bancos de dados NoSQL, o TimescaleDB® apresenta algumas vantagens significativas. Dentre elas, podemos citar a manutenção da linguagem SQL, a vantagem de se trabalhar com dados de séries temporal e dados relacionais em um mesmo banco de dados, tornando possível a realização de consultas envolvendo a junção de tabelas de série temporal e relacionais, além da implementação de ferramentas desenvolvidas por terceiros.

4 CONCLUSÃO

Neste trabalho, foi possível constatar a importância das tecnologias de banco de dados no contexto da Indústria 4.0, que por vezes, não recebe o mesmo destaque que as tecnologias de disponibilização de dados através de dispositivos inteligentes, as redes industriais, e as tecnologias de processamento de dados, com linguagens de programação que utilizam conceitos de IA.

Porém, quando falamos em histórico de dados provenientes de processos industriais, uma análise ou estudo mais aprofundado é recomendado para a escolha da tecnologia de banco de dados mais adequada. Quando pensamos nos populares bancos de dados relacionais que utilizam a linguagem SQL, estes são recomendáveis para o armazenamento de dados que não possuem como característica manter um histórico de série temporal, como por exemplo ordens de produção, receitas, *setups,* dados de produtos e etc. Entretanto, estes tipos de banco de dados não trabalham bem com dados históricos de série temporal de processos industriais, como por exemplo, dados de temperatura ou pressão, que dependendo do processo, para se fazer uma análise satisfatória, exige-se que os dados destes sensores sejam armazenados, por exemplo, a cada 1 segundo ou menos.

Além disso, considerando a característica dos bancos de dados relacionais trabalhar com dados estruturados, ou seja, dados em tabelas, pode ser tornar uma tarefa difícil ao generalizar em uma única tabela todos sensores e dispositivos de uma planta industrial, mesmo porque as chances destes sensores e dispositivos possuírem métricas (atributos) diferentes são grandes. Desta forma, a falta de flexibilidade dos esquemas de bancos de dados relacionais se torna um ponto negativo quando se considera a possibilidade de trabalhar com dados de série temporal em bancos de dados relacionais.

Como alternativa a este problema, surgem os bancos de dados NoSQL, que possuem como principal característica não serem relacionais, trabalhar com dados não-estruturados, ausência de esquemas ou esquemas flexíveis, além de possuir uma arquitetura desenvolvida para ingerir altas taxas de dados, serem altamente escaláveis graças a sua capacidade de trabalhar de forma distribuída, em *clustering.*

Com as duas tecnologias apresentadas anteriormente, podemos considerar a possibilidade de suprir as demandas de armazenamento de dados industriais, podendo-se utilizar os bancos de dados relacionais para armazenar dados transacionais ou relacionais, e bancos NoSQL para armazenar dados de série temporal. Porém esta solução exige trabalhar com duas tecnologias bem diferentes, com linguagens e estruturas diferentes quando comparadas entre si. Isto exige mais conhecimentos e investimentos na administração de manutenção destes dados.

Porém os bancos de dados de série temporal surgem como uma terceira via para se trabalhar com dados industriais, falando mais especificamente do TimeserialDB®. Implementado sobre o PostgreSQL®, o TimeserialDB® fornece funcionalidades para trabalhar com dados relacionais, com linguagem SQL, como qualquer banco de dados de natureza relacional e a flexibilidade de se trabalhar com dados de série temporal em uma mesma plataforma, eliminando problemas causados pelo fato de se trabalhar com bancos de dados diferentes em um mesmo sistema, e dispensa a existência de uma equipe de profissionais com conhecimento em multiplataformas, uma vez que dados de naturezas tão diferentes podem ser armazenados em um único banco de dados.

Como trabalho futuro, poderia ser realizado um estudo de caso utilizando o TimeserialDB® com a finalidade de se analisar em uma aplicação práticas as vantagens trazidas por este banco de dados.

REFERÊNCIAS

BORTOLINI, Marco et al. Assembly system design in the Industry 4.0 era: a general framework. **IFAC-PapersOnLine**, v. 50, n. 1, p. 5700-5705, 2017. DOI 10.1016/j.ifacol.2017.08.1121. Disponível em: https://www.researchgate.net/publication/320496225_Assembly_system_design_in_the_Industry_40_era_a_general_framework. Acesso em: 10 mai. 2019.

BRYNJOFSSON, Erik; McAFEE, Andrew. **A segunda era das máquinas:** trabalho, progresso e prosperidade em uma época de tecnologias brilhantes. Rio de Janeiro: Alta Books, 2015. 338 p.

CASSOL, Tiago Sperb. **Um estudo sobre alternativas de representação de dados temporais em bancos de dados relacionais**. 2012. 105 f. Dissertação (Mestrado em Computação) – Universidade Federal do Rio Grande do Sul. Porto Alegre, 2012. Disponível em: http://hdl.handle.net/10183/67849. Acesso em: 10 mai. 2019.

COELHO, Pedro Miguel Nogueira. **Rumo à indústria 4.0**. 2016. 65 f. Dissertação (Mestrado em Engenharia Mecânica). Universidade de Coimbra – Faculdade de Ciências e Tecnologia. Coimbra, 2016. Disponível em: http://hdl.handle.net/10316/36992. Acesso em: 13 mai. 2019.

DB-ENGINES. **DB-Engines Ranking.** c2019. Disponível em: https://db-engines.com/en/ranking. Acesso em: 13 mai. 2019.

GOULART, Victor. Sistema IoT para monitoramento de porteiras utilizando LoRa e LoRaWAN. In: CIMATech, 5, São Jose dos Campos-SP, 2018. **Anais eletrônico [...]**. São José dos Campos: FATEC, 2018. Disponível em:

https://publicacao.cimatech.com.br/index.php/cimatech/article/view/108. Acesso em: 13 mai. 2019.

GREENFIELD, David. **Manufacturing Databases Explained.** 2012**.** Disponível em: https://www.automationworld.com/article/technologies/databases-historians/manufacturing-databases-explained. Acesso em: 13 mai. 2019

LOUREIRO, Rodrigo. Os dados são o novo petróleo. **Isto é Dinheiro**, São Paulo, n. 1060, 9 mar. 2018. Disponível em https://www.istoedinheiro.com.br/os-dados-sao-o-novo-petroleo/. Acesso em: 10 mai. 2019.

LÓSCIO, Bernadette Farias; OLIVEIRA, H. R de; PONTES, J. C de S. NoSQL no desenvolvimento de aplicações Web colaborativas. In: Simpósio Brasileiro de Sistemas Colaborativos, 8. 2011. Rio de Janeiro. **Anais** [...]. Rio de Janeiro: SBC, 2011. p. 11.

MADAKAM, S., RAMASWAMY, R.; TRIPATHI, S. Internet of Things (IoT): a Literature Review. **Journal of Computer and Communications**, 3, 164 -173, 2015. Doi 10.4236/jcc.2015.35021.Disponivel em: https://www.researchgate.net/publication/280527542_Internet_of_Things_IoT_A_Literature_Review . Acesso em: 13 mai. 2019.

MATOS, David. **Top 6 NoSQL Databases**. 2018. Disponível em: http://www.cienciaedados.com/top-6-nosql-databases/. Acesso em: 20 mai. 2019.

MONGODB. **What is MongoDB**? [2019]. Disponível em: https://www.mongodb.com/what-is-mongodb. Acesso em 20 mai. 2019

PORTAL LUBES. **Pacote para indústria 4.0 vai ser lançado em março.** 2018. Disponível em: http://portallubes.com.br/2018/02/pacote-para-industria-4-0/. Acesso em: 01 ago. 2019

QUEIROS, Maycon P. **Introdução ao Cassandra.** 2017. Disponível em: https://www.devmedia.com.br/introducao-ao-cassandra/38377. Acesso em: 26 mai. 2019

SILBERSCHATZ, Abraham; KORTH, Henry F.; SUDARSHAN, S. **Sistema de banco de dados.** 2012. 6. ed. Elsevier. 861 p.

TEOREY, Toby et al. **Projeto e modelagem de bancos de dados**. ed. Rio de Janeiro: Elsevier, 2014. 328 p.

TIMESCALE. **TimescaleDB Documentation**. 2019. Disponível em: https://docs.timescale.com/v1.3/main. Acesso em 26 mai. 2019.

TULIP**. The Ultimate Guide to Industrial IoT for Manufacturers.** 2018. E-book. Disponível em: https://tulip.co/resources/iiot-for-manufacturers/. Acesso em: 01 ago. 2019.

Sobre os autores:

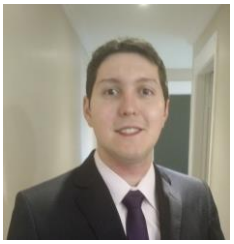
[i] **João Bosco Pedralino Silva**

Possui graduação em Sistemas de Informação, cursando atualmente a Pós Graduação em Indústria 4.0 pela Faculdade SENAI de Tecnologia Mecatrônica (2019). Possui experiência na área de Engenharia de Automação e Controle, com ênfase em projetos e programação de máquinas e sistemas. É pesquisador no departamento de P&D na empresa Nadir Figueiredo, responsável por pesquisas de tecnologias digitais.
CV:

[ii] **Daniel Otavio Tambasco Bruno**

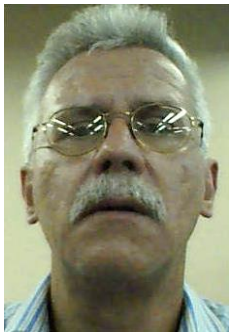
Doutorando e Mestre em Engenharia da Informação pela Universidade Federal do ABC (2013). Especialista em Análise, desenvolvimento de Sistemas e Banco de Dados pela Universidade de Ribeirão Preto (2007), Especialista em Educação a Distância pela Universidade Paulista (2012). Bacharel em Análise de Sistemas pela Universidade Paulista (2003). Atualmente é Técnico em Manufatura Digital e Professor da Faculdade SENAI de Tecnologia Mecatrônica Industrial. Tem experiência na área de Inteligência Artificial, Internet das coisas, Gestão de Tecnologia da Informação e Desenvolvimento de Sistemas de Informação.
CV: http://lattes.cnpq.br/3491851270517427

[iii] **Thiago Tadeu Amici**



Ministra aulas na pós-graduação de Indústria 4.0, na graduação em Tecnologia em Mecatrônica Industrial e no técnico em Mecatrônica no SENAI Armando de Arruda Pereira, além de assessorar o Instituto SENAI de Tecnologia Metalmecânica. 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. 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. CV: http://lattes.cnpq.br/9165856219131658

[ii] **Paulo Sebastião Ladivez**



Possui graduação em Engenharia Elétrica pela Universidade Mogi das Cruzes (1984) com especialização em Tecnologias e Sistemas de Informação pela Universidade Federal do ABC (2013). Atualmente é professor da Faculdade SENAI de Tecnologia Mecatrônica, lecionando as disciplinas Projetos, Microcontroladores, Linguagem de Programação no curso Tecnológico em Mecatrônica Industrial e na Pós-Graduação em Automação Industrial. Tem experiência na área de Engenharia Eletrônica, com ênfase em Automação Industrial e Mecatrônica, atuando principalmente nos seguintes temas: Mecatrônica, Manufatura Digital, Redes Industriais, Automação Industrial, Microcontroladores e Controle.
CV: http://lattes.cnpq.br/7235073442326291