

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
CURSO DE CIÊNCIA DA COMPUTAÇÃO

ANA CAROLINA SANCHOTENE SILVA

**Análise de Uso de Banco de Dados por
Sistemas de Automação Industrial**

Monografia apresentada como requisito parcial
para a obtenção do grau de Bacharel em Ciência
da Computação

Orientador: Prof. Dr. Érika Cota
Co-orientador: Prof. Dr. João Netto

Porto Alegre
2018

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Reitor: Prof. Rui Vicente Oppermann

Vice-Reitora: Prof^a. Jane Fraga Tutikian

Pró-Reitor de Graduação: Prof. Wladimir Pinheiro do Nascimento

Diretora do Instituto de Informática: Prof^a. Carla Maria Dal Sasso Freitas

Coordenador do Curso de Ciência da Computação: Prof. Sérgio Luis Cechin

Bibliotecária-chefe do Instituto de Informática: Beatriz Regina Bastos Haro

"A sorte ajuda quem se prepara"
— EDNA MODE - OS INCRÍVEIS

AGRADECIMENTOS

Agradeço aos meus pais pelos conselhos, carinho e apoio, ao meu irmão pelos abraços sempre que eu precisei. Aos meus amigos pela compreensão e pela ajuda nos momentos de tensão.

Agradeço à professora Érika pela paciente orientação, ao Ivan Elisandro da Silva por todas as dicas, material e tempo disponibilizado, ao professor Netto pelas correções realizadas e à banca que dedicou seu tempo à leitura desse trabalho.

Aos professores Taisy Weber e Raul Fernando Weber, *in memoriam*, por todo carinho nas minhas primeiras semanas na computação.

RESUMO

O presente trabalho tem como objetivo a realização de um teste de performance em um *software* de automação industrial, o sistema SCADA *BlueWave*. Dadas informações de clientes insatisfeitos com o sistema porque utiliza mais espaço em disco que seus concorrentes, foi realizada uma avaliação que visa encontrar a causa disto. Para tal, é feita a definição do contexto de automação industrial bem como os passos para criação de um plano de teste de performance que compara seu desempenho com o de um concorrente, o *Ignition*. Os resultados, apesar de refutarem as hipóteses levantadas, permitiram avançar na solução do problema, visto que uma falha no sistema foi detectada.

Palavras-chave: Automação Industrial. BlueWave. Ignition. Desempenho. SCADA.

Analysis of Database Usage by Industrial Automation Systems

ABSTRACT

The present work aims to execute a performance test in an industrial automation software, the SCADA system BlueWave. Given information from customers that were dissatisfied with the product because it uses more disk space than its competitors, an evaluation was made to find the cause of this. To accomplish this, the definition of how the automation systems work as well as the steps to create a performance test plan that compares its performance with that of a competitor, (Ignition system) are described and followed. The results, while refuting the hypotheses raised, bring us closer to solving the problem, since a defect in the system was found.

Keywords: Industrial Automation. BlueWave. Ignition. Performance. SCADA.

LISTA DE FIGURAS

Figura 2.1	Fonte: autora. Representação da pirâmide de hierarquia da indústria de automação, os sistemas são complementares.....	20
Figura 2.2	Fonte: (SYSTEMS, 2005). Exemplo de uma tabela representando a qualidade da informação em um historiador.....	28
Figura 2.3	Fonte: autora. Representação de uma rede de aquisição. Nessa figura, o historiador está inserido na categoria SCADA.	29
Figura 2.4	Fonte: Documentação dos desenvolvedores do BlueWave. Fluxograma representando como as classes são chamadas dentro do historiador, começando pela chamada externa ao mesmo.....	31
Figura 2.5	Fonte: Documentação dos desenvolvedores do BlueWave. Diagrama de classes historiador do BlueWave.....	32
Figura 2.6	Fonte: Documentação dos desenvolvedores do BlueWave. Diagrama de módulos da integração do historiador do BlueWave com outros componentes do sistema.....	33
Figura 2.7	Fonte: (AUTOMATION, 2018c). Imagem ilustrativa da arquitetura do <i>Ignition</i> utilizada nesse trabalho.	35
Figura 3.1	Relatório de uso de espaço em disco (em Kb) após os testes de múltiplas inserções.....	45
Figura 3.2	Relatório das tabelas do BlueWave durante a preparação do banco de dados	46
Figura 3.3	Análise da tabela de banco de dados do BlueWave.....	46
Figura 3.4	Análise da tabela de banco de dados do BlueWave.....	46
Figura A.1	Uso de disco das tabelas do Ignition antes do início dos testes.....	52
Figura A.2	Uso de disco das tabelas do BlueWave antes do início dos testes.....	52
Figura A.3	Uso de disco das tabelas do Ignition após uma tag ter seu valor modificado	52
Figura A.4	Uso de disco das tabelas do BlueWave após uma tag ter seu valor modificado.....	52
Figura B.1	Uso de disco das tabelas do Ignition após todas as tags terem seu valor modificado.....	53
Figura B.2	Uso de disco das tabelas do BlueWave após todas as tags terem seu valor modificado.....	53
Figura C.1	Uso de disco das tabelas do BlueWave antes de todas as tags terem seu valor modificado diversas vezes.....	54
Figura C.2	Uso de disco das tabelas do Ignition antes de todas as tags terem seu valor modificado diversas vezes.....	54
Figura C.3	Uso de disco das tabelas do BlueWave após todas as tags terem seu valor modificado diversas vezes - Rodada 1.....	55
Figura C.4	Uso de disco das tabelas do Ignition após todas as tags terem seu valor modificado diversas vezes - Rodada 1.....	55
Figura C.5	Uso de disco das tabelas do BlueWave após todas as tags terem seu valor modificado diversas vezes - Rodada 2.....	55

Figura C.6 Uso de disco das tabelas do Ignition após todas as tags terem seu valor modificado diversas vezes - Rodada 2	55
Figura C.7 Uso de disco das tabelas do BlueWave após todas as tags terem seu valor modificado diversas vezes - Rodada 3	56
Figura C.8 Uso de disco das tabelas do Ignition após todas as tags terem seu valor modificado diversas vezes - Rodada 3	56

LISTA DE TABELAS

Tabela 3.1	Tabela representando a estrutura de dados criada pelo <i>Ignition</i>	42
Tabela 3.2	Tabela representando a estrutura de dados criada pelo <i>BlueWave</i>	43
Tabela D.1	Tabela explicitando uso de disco dos historiadores <i>BlueWave</i> e <i>Ignition</i> em cada uma das rodadas dos testes de longa duração(em Kb)	57

LISTA DE ABREVIATURAS E SIGLAS

BD	Banco de Dados
BW	BlueWave
DCS	Distributed Control System
ERP	Enterprise Resource Planning
Kb	Kilobit
PIMS	Plant Information Management System
PLC	Programmable Logic Controller
RTU	Remote Terminal Unit
SCADA	Supervisory Control and Data Acquisition
SQL	Structured Query Language
TI	Tecnologia da Informação

SUMÁRIO

1 INTRODUÇÃO	13
1.1 Objetivo	14
1.2 Estrutura do Trabalho	14
2 FUNDAMENTAÇÃO TEÓRICA	16
2.1 Teste de Desempenho	16
2.2 Indústria de Automação	19
2.2.1 SCADA	21
2.2.2 PIMS	23
2.2.3 DCS.....	24
2.2.4 ERP	24
2.3 Logs	25
2.4 Historiadores	27
2.5 BlueWave	30
2.6 Ignition	33
3 DESENVOLVIMENTO	36
3.1 Definição dos objetivos do estudo e os limites do sistema	36
3.2 Definição dos serviços do sistema e os possíveis resultados	37
3.3 Seleção de métricas de desempenho	37
3.4 Definição dos parâmetros do sistema e do workload	38
3.5 Seleção dos fatores e seus valores	39
3.6 Seleção técnicas de avaliação	39
3.6.1 Primeira Hipótese.....	40
3.6.2 Segunda Hipótese.....	40
3.7 Seleção do workload	40
3.7.1 Primeira Hipótese.....	40
3.7.2 Segunda Hipótese.....	41
3.8 Projeção dos experimentos	41
3.8.1 Primeira Hipótese.....	41
3.8.2 Segunda Hipótese.....	42
3.9 Análise e interpretação dos dados	42
3.9.1 Primeira Hipótese.....	42
3.9.2 Segunda Hipótese.....	44
3.9.2.1 Atualização de um valor da tabela	44
3.9.2.2 Atualização de todos os valores da tabela uma vez	44
3.9.2.3 Atualização de todos os valores da tabela diversas vezes.....	44
3.10 Apresentação dos resultados	45
4 CONCLUSÃO	47
4.1 Trabalhos Futuros	48
REFERÊNCIAS	49
APÊNDICE A — RELATÓRIO GERADO PELO MICROSOFT SQL SERVER 2014 EXPLICITANDO O USO DE DISCO POR CADA TABELA NO ESTADO INICIAL E APÓS A INSERÇÃO DE UM ITEM	52
APÊNDICE B — RELATÓRIO GERADO PELO MICROSOFT SQL SERVER 2014 EXPLICITANDO O USO DE DISCO POR CADA TABELA APÓS A MODIFICAÇÃO DE TODOS OS PONTOS DE COLETA UMA VEZ..	53
APÊNDICE C — RELATÓRIO GERADO PELO MICROSOFT SQL SERVER 2014 EXPLICITANDO O USO DE DISCO POR CADA TABELA APÓS OS TESTES DE LONGA DURAÇÃO	54

**APÊNDICE D — TABELA COM VALORES EM KB DESCRIVENDO ESPAÇO
DE USO EM DISCO APÓS OS TESTES DE LONGA DURAÇÃO57**

1 INTRODUÇÃO

O presente trabalho traz um estudo dos historiadores de dois sistemas diferentes a fim de entender por que um deles está em baixo conceito com seus clientes, que indicam que o mesmo faz um uso de disco muito maior que seus concorrentes¹. São feitas avaliações tanto da forma como as tabelas são criadas no banco de dados a partir das informações inseridas no sistema quanto uma avaliação de como as informações armazenadas ocupam espaço buscando compreender a insatisfação de seus clientes.

A automação está cada vez mais presente no dia-a-dia de todos e a indústria é um segmento que vem, progressivamente, abrindo mão de processos manuais para substituí-los por máquinas, tornando a produção mais precisa e barata. Com o objetivo de melhorar o resultado final da manufatura, são utilizados sistemas de automação de processos. Como será apresentado nesse trabalho, a automação industrial se desenvolveu como uma hierarquia, na qual o *software* que está na base da mesma é um sistema que lida com grandes quantidades de dados, utilizando um historiador para fazê-lo.

Historiadores são um tipo de *logger*. *Logs* são utilizados para, através de dados coletados durante a execução do sistema, que seja possível determinar possíveis problemas e a causa dos mesmos. Mais informações podem ser encontradas em (CORPORATION, 2018b) (APACHE, 2018) e os *loggers* são os *softwares* que fazem essa coleta. Esses sistemas são específicos para coleta e armazenamento de dados de “chão de fábrica” e têm uma forma de coleta de dados diferenciada, utilizando dados que são selecionados pelos usuários. Outra diferença é que cada informação coletada também é salva com dados sobre data, que possibilitam saber a qualidade da coleta e o momento em que foi efetuada. Historiadores geralmente estão presentes dentro de um Sistema de Supervisão e Aquisição de Dados, também conhecido como SCADA pelo seu nome em inglês "*Supervisory Control And Data Acquisition*") (EXCELENCE, 2018). Um sistema SCADA é um dos componentes de um sistema de automação de uma planta e tem como principal responsabilidade a coleta e o armazenamento de dados.

No contexto do chão de fábrica, geralmente um número alto de dados é coletado e o sistema deve fazer essa coleta de forma que não haja perda de informações, o que pode consumir os recursos de forma rápida. É necessário que o historiador execute com recursos o suficiente para que seja capaz de manter o estado do sistema condizente com o estado real da planta. Também é necessário que o mesmo tenha capacidade

¹Informação recebida pelos responsáveis pelo sistema

para processar constantemente e corretamente quando as coletas devem ser efetuadas e atualizar as tabelas. Como todos os processos estão interligados, a falta de escalabilidade de um componente pode significar que todo o sistema ficará lento ou incapaz de realizar a tarefa proposta.

Para que todos os sistemas presentes na hierarquia tenham um bom desempenho e reflitam o real estado de todo o chão de fábrica, é preciso que o historiador seja rápido, robusto e capaz de armazenar uma grande quantidade de dados sem perder eventos. O objeto de estudo desse trabalho é o sistema *BlueWave*(AUTOMAÇÃO, 2012b) ou BW, desenvolvido pela Altus, empresa que tem parceria com a Universidade Federal do Rio Grande do Sul².

Para a realização do presente trabalho, foi utilizada a metodologia de medição, que consiste na execução do sistema, coleta de métricas e, então, uma comparação com a *baseline*³. As métricas utilizadas serão os relatórios gerados pelo *SQL Server 2014*, o sistema de BD utilizado neste estudo. Esses relatórios trazem, em Kilobits, ou Kb, o espaço em disco utilizado por cada tabela no banco de dados. Para que seja feita a validação, os testes foram realizados também em outro historiador, presente no sistema *Ignition*.

1.1 Objetivo

O objetivo desse trabalho é, em um estudo de caso específico, avaliar o comportamento de bancos de dados de um sistema SCADA, bem como entender como é feito o uso de espaço em disco do *BlueWave* e avaliar seu desempenho quando comparado com outro *software* da indústria. Para essa comparação, será feita a análise do espaço de disco utilizado.

1.2 Estrutura do Trabalho

O trabalho está estruturado em quatro capítulos além desta introdução. No Capítulo 2 são aprofundados os conceitos que foram utilizados na realização da pesquisa, tendo

²Esse projeto é uma parceria público-privada na qual há um financiador público, nesse caso o BNDES, e uma contrapartida privada, no caso a Altus. O objetivo deste projeto é produzir *hardware* e *software* para a indústria de automação, ou seja, o projeto tem como saída produtos que vão efetivamente para o mercado.

³Uma *baseline* é uma medição feita em outro sistema ou mesmo uma versão anterior do *software* alvo para fazer a validação do teste

como objetivo preparar o leitor para compreender a parte prática do mesmo. O Capítulo 3 contém as definições de como foram realizados os testes e suas especificações técnicas e, finalmente, o Capítulo 4 apresenta a análise a partir dos resultados obtidos, discutindo erros e acertos das escolhas tomadas durante o processo. O Capítulo 5 conclui o trabalho e discute possíveis estudos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

O foco principal desse trabalho são os historiadores nos sistemas de automação industrial. Portanto, é importante entender o contexto nos quais os mesmos estão inseridos e o impacto de sua performance.

No presente capítulo são apresentados os conceitos que fundamentam teoricamente o estudo e as principais características de *softwares* que fazem parte da hierarquia de automação industrial. Esse capítulo se divide em duas partes: na primeira parte, são definidos os conceitos e passos para a criação de teste de desempenho e análise dos resultados (Seção 2.1). Na segunda parte, estão definidos conceitos da indústria de automação (Seção 2.2) e diversos sistemas que fazem parte desse escopo (Seções 2.3 e 2.4) e, então, são apresentadas as arquiteturas dos dois sistemas que são objeto de análise desse trabalho, o *BlueWave* (Seção 2.5) e o *Ignition* (Seção 2.6);

2.1 Teste de Desempenho

Testes de desempenho são criados a partir de processos complexos que exigem um profundo conhecimento de como o sistema a ser testado funciona e como o mesmo é utilizado por seus usuários. Para iniciar uma avaliação de performance é necessária uma abordagem sistemática que começa com a definição de quais são os objetivos do teste. (JAIN, 1991) propõe nove passos para concepção, realização e apresentação de resultados de uma avaliação de performance. São elas:

- Definir os objetivos do estudo e os limites do sistema.
- Listar os serviços do sistema e os possíveis resultados.
- Selecionar métricas de desempenho.
- Listar os parâmetros do sistema e do *workload*.
- Selecionar fatores e seus valores.
- Selecionar técnicas de avaliação.
- Selecionar o *workload*
- Projetar os experimentos.
- Analisar e interpretar os dados.
- Apresentar os resultados. Começar de novo, se necessário.

O escopo do teste diz quais são os limites do sistema. Por exemplo, se o sistema de login de uma página na internet tem uma chamada para o *Google Analytics*, é interessante retirar essa chamada do teste de performance do login, já que o tempo de resposta dessa chamada está fora do controle. O propósito do teste está diretamente ligado à decisão de como será executada a avaliação, como veremos adiante. Para que os resultados obtidos pelos testes sejam relevantes para a avaliação da aplicação, é necessário conhecimento do ambiente no qual a mesma será utilizada e tentar replicá-lo (JAIN, 1991).

É necessário escolher quais serão as métricas utilizadas para avaliação e o *workload*. *Workload*¹ pode ser definido como a quantidade de “trabalho” que o sistema terá. Para o escopo desse trabalho, o workload também será chamado de suite de teste. A definição dos casos de testes para uma avaliação de desempenho é uma atividade que exige conhecimento da aplicação, compreensão de qual será o contexto do usuário que a utiliza e suas necessidades e expectativas.

Um passo fundamental no processo de definição de escopo é enumerar os serviços que estão dentro do escopo do teste e suas possíveis saídas e, a partir disso, decidir qual a melhor forma de realizar a análise. A performance de um sistema pode ser medida pelo seu tempo de resposta, sua velocidade de processamento, sua taxa de erros, entre outros critérios. Também é necessário definir se será feito um estudo para avaliar a experiência do usuário, avaliando itens como disponibilidade, ou a saúde geral do sistema, através de variáveis como a utilização de recursos. É importante que o conjunto de métricas definidas atenda o objetivo do teste e represente todos os limites do sistema estabelecidos.

Existem dois tipos de parâmetros que influenciam no desempenho do sistema:

- *Parâmetros do sistema*: características de *software*, como a forma que a paginação do banco de dados, ou *hardware*, como a quantidade de memória dos servidores.
- *Parâmetros de carga*: nesse escopo estão as características ditadas por como o usuário final usa o sistema, quantidade de usuários, etc.

É preciso decidir quais são os parâmetros que impactam o sistema e, então, selecionar quais serão testados. Para isso, é preciso, primeiramente, definir quais são os fatores que mais afetam a performance através da avaliação dos serviços selecionados.

Existem três técnicas diferentes para avaliação de desempenho (JAIN, 1991). São elas:

- *Modelagem Analítica*: esse modelo de avaliação parte de uma abstração do sistema

¹ Mais informações sobre workload podem ser encontradas em (CORPORATION, 2018d)

que tem como objetivo prever o comportamento do mesmo. A modelagem analítica permite "a formulação de equações através das quais o comportamento do sistema pode ser obtido"(COLCHER, 2009). Das alternativas apresentadas, essa é a mais barata, mas também a que tem menor confiabilidade, pois exige uma grande quantidade de simplificações e suposições sobre do objeto de teste.

- *Simulação*: a simulação parte de uma abstração algorítmica do *software*. É necessário decidir quais parâmetros serão estudados para que se possa começar a realizar esses testes. Eles têm um alto custo computacional, pois precisam de um grande número de rodadas para definir se os resultados são confiáveis (CIGNO, 2017).
- *Medição*: essa técnica de avaliação só é possível se algo similar ao objeto de estudo já existe como, por exemplo, quando a nova versão de um sistema é testada, pois implica em realizar um teste com as mesmas configurações em duas versões diferentes do sistema para determinar se houve melhora, piora ou não houve impacto no escopo testado. Esse método tem outras limitações como tempo para realizar a atividade, já que algumas medições podem demorar, o alto custo desse formato de teste, entre outros.

Dentre as técnicas de avaliação disponíveis, diferentes métodos de definição de carga² podem ser utilizados. Sua escolha depende do escopo do teste. O método utilizado para este trabalho é *benchmark* de aplicação. *Benchmark* significa "um padrão ou ponto de referência contra o qual as coisas podem ser comparadas" e, mais especificamente no âmbito de medição de performance, "um teste projetado para avaliar ou comparar o desempenho de hardware ou software de computador"(HUNTER, 1982).

Existem duas formas de avaliar através dessa categoria de carga:

- *Benchmark da indústria*: experimentos desse tipo podem ser feitos de duas maneiras:
 - *i)* a partir de *softwares* de monitoramento, como o Dynatrace, coletam métricas dos sistemas que o utilizam para que seja possível determinar se a performance atual está de acordo com a média da indústria (LLC, 2017);
 - *ii)* definir casos de testes nos quais dois sistemas da mesma indústria sejam testados em um ambiente idêntico para que seja possível comparar o resultado.
- *Benchmark do próprio sistema*: testes realizados dessa forma exigem que seja feita uma medição de uma versão anterior do sistema (conhecida como *baseline*) para

²Mais informações a respeito dos diferentes métodos de definição de carga podem ser encontrados em (JAIN, 1991).

que seja possível a comparação com o resultado da avaliação.

Esse trabalho está baseado no modelo de *benchmark* da indústria.

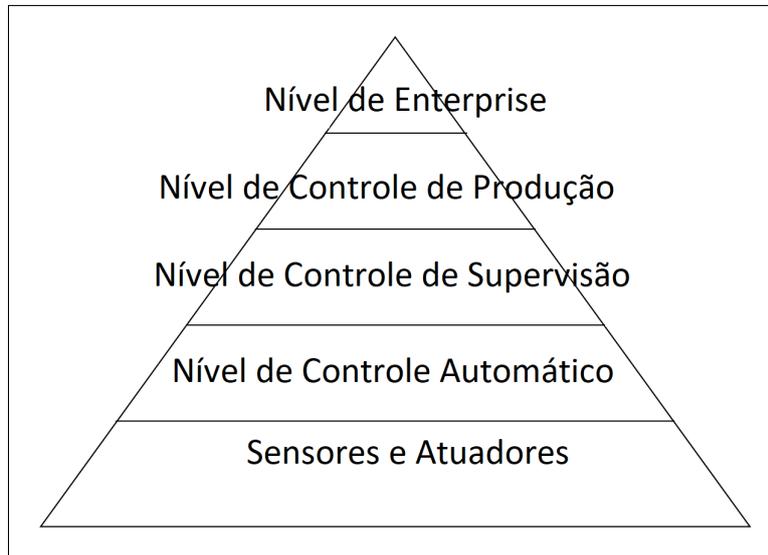
Os dados de um teste de *software* de performance podem ser interpretados utilizando diferentes abordagens. Por exemplo, é possível que um sistema X lide melhor com a falta de memória que um sistema Y porém, o sistema Y desempenha melhor em todos os outros cenários. Assim, é possível que a empresa que desenvolveu o sistema X anuncie que ele é melhor que seu concorrente, omitindo a realidade sobre o mesmo.

Após os experimentos, faz-se a análise dos dados. Essa análise será probabilística, pois o sistema está sendo avaliado a partir de um número finito de amostras e essas não necessariamente cobrem todos os casos de uso reais. Por exemplo, é possível que um usuário utilize o sistema de forma incorreta, o que leva a um uso não previsto e pode ocasionar em problemas de performance. É preciso ter cuidado ao apresentar os resultados de uma análise de desempenho. É necessário que seja levada em consideração a facilidade de leitura para o público ao qual o estudo está sendo apresentando. Os interessados na performance de um sistema nem sempre são da área da tecnologia e, assim, é necessário que a conclusão da avaliação seja clara.

2.2 Indústria de Automação

A automação industrial pode ser definida como "o uso de tecnologias de um conjunto de dispositivos de controle automático que resultam na operação e controle automáticos de processos industriais sem intervenção humana significativa e alcançando desempenho superior ao controle manual"(TECHNOLOGY, 2015). O desempenho superior citado acima pode se referir a produtos de maior qualidade, uma produção mais eficiente, maior segurança do trabalho, entre outros. Um sistema de automação industrial tem diferentes níveis(M.M., 2014), como mostrado na Figura 2.1:

Figura 2.1: Fonte: autora. Representação da pirâmide de hierarquia da indústria de automação, os sistemas são complementares.



- *Sensores e Atuadores*: esse é o nível mais baixo da hierarquia da automação e atua diretamente na produção. É formado por sensores (medidores de temperatura e pressão, por exemplo) e atuadores (que são capazes de realizar ações como aumentar o resfriamento de uma caldeira).
- *Nível de Controle Automático*: capaz de se comunicar com toda a planta, o nível de controle é implementado com o uso de *Programmable Logic Controllers* (PLCs), ou Controladores Lógicos Programáveis, e *Remote Terminal Units* (RTUs), ou Unidades Terminais Remotas. Sua responsabilidade é coletar os dados utilizando os sensores e responder aos mesmos a partir de rotinas de controle, enviando comandos aos atuadores.
- *Nível de Controle de Supervisão*: nessa camada estão sistemas como SCADA (definido na Seção 2.2.1), DCS (definido na Seção 2.2.3), entre outros, cujo objetivo é a supervisão dos processos que estão ocorrendo na planta. Esse nível também é responsável pelo monitoramento dos pontos de coleta e armazenamento histórico. Aqui entram os historiadores (definidos na Seção 2.4) e configurações gerais da planta.
- *Nível de Controle de Produção*: nesse escopo estão os responsáveis pela manutenção, qualidade, produção e inventário como o Sistema de Gerenciamento de Informação de Processo ou *Process Information Management System*, também conhecidos como PIMS (definido na Seção 2.2.2).
- *Nível de Enterprise*: esse nível gerencia todo o sistema de automação. Suas atribuições

são planejamento de produção, análise de cliente, ordens de compras, etc. Essa camada está mais ligada ao nível administrativo, influenciando na tomada de decisões. Aqui se encontram *softwares* como o ERP, que também é conhecido como *Enterprise Resource Planning* ou Planejamento de recurso corporativo (definido na Seção 2.2.4).

Os *softwares* de um sistema de automação podem ser operados por três tipos diferentes de usuários:

- *Engenheiro de Processo*: esse profissional tem como objetivo melhorar o desempenho financeiro e produtivo de uma indústria através do estudo das atividades de produção e melhoria nos processos, equipamentos, uso de mão de obra e matéria-prima. No contexto de automação, ele é responsável pela criação e interpretação da representação da planta no sistema, criação de alarmes, correlacionamento de informações, entre outras demandas.
- *Técnico de Operação*: é dever deste funcionário desenvolver relatórios, análises, fazer o acompanhamento e a tomada de decisão em relação aos pontos monitorados pelo sistema.
- *Técnico de Manutenção*: entre outras atribuições, ele é responsável por atender as demandas informadas pelos alarmes do sistema de automação. O técnico deve realizar o suporte de manutenção nas áreas de produção, visando atender os reparos nos equipamentos industriais em apoio às equipes de eletromecânicos.

Como pode ser visto, cada nível tem suas características e responsabilidades, criando um fluxo de informação do chão de fábrica em direção ao nível de *enterprise* e o fluxo contrário provendo instruções sobre o processo a ser seguido. Sendo assim, esses sistemas são complementares.

Os principais sistemas que fazem parte dessa hierarquia são descritos na sequência.

2.2.1 SCADA

Os SCADA são sistemas de controle, supervisão e administração de alto nível (EXCELENCE, 2018). SCADA significa *Supervisory Control and Data Acquisition* ou Controle Supervisório e Aquisição de Dados em tradução livre. Eles utilizam protocolos de comunicação industrial nas diferentes interfaces dos periféricos com os quais o sistema é integrado. Essas interfaces são as que possibilitam o monitoramento e gerenciamento

das atividades que são supervisionadas através do SCADA.

Uma das principais características destes sistemas é a possibilidade de guardar toda a informação "recolhida" pelos sensores em um único *software* e a capacidade de enviar comandos para diferentes RTUs e PLCs. Os PLCs são utilizados em processos industriais, frequentemente de menor tamanho, e disponibilizam uma grande variedade de protocolos específicos para esses processos. As RTUs são derivadas dos PLCs, mas mais voltadas para a automação nos setores de geração e distribuição de energia elétrica. Elas têm maior robustez eletromagnética e mecânica, em função dos ambientes com maior risco de interferência nos quais são instaladas. Em alguns casos, RTUs possuem protocolos presentes em processos industriais, mas são mais focados em protocolos desenvolvidos para área elétrica.

Através do SCADA, é possível acessar informações sobre todo o sistema automatizado de forma gráfica, alarmes e os logs da planta. O sistema pode gerar diferentes formas de relatório e gerenciar as notificações do sistema a partir de uma configuração previamente definida pelo engenheiro de projeto. A interface entre o sistema e os engenheiros possibilita que estes sejam capazes de fazer modificações aos controladores em tempo real, tarefa que pode ser auxiliada pelos alarmes.

Os alarmes são utilizados para informar que um sistema saiu de um estado otimizado de operação e que uma ação externa (de um engenheiro ou operador) é necessária para que a planta retorne para aquele estado. Por exemplo, quando é necessária alguma intervenção manual, como trocar o filtro de uma das saídas de ar, o sistema é capaz de enviar um alarme informando sobre a necessidade de troca. Essas notificações padronizadas focam primariamente no que deve ser feito, não nos passos que devem ser seguidos caso o alarme seja acionado, o que as torna padronizáveis mesmo que sistemas SCADA estejam presentes em diversas indústrias diferentes. É recomendado que os alarmes criados sigam a norma ANSI/ISA 18.2 (HOLLIFIELD, 2010), que tem como objetivo melhorar a segurança dos sistemas.

Um sistema SCADA se comunica com o chão de fábrica de duas formas: através da comunicação serial, como o *Modbus*, ou da comunicação *ethernet*, como a fibra óptica, (CORPORATION, 2012). A comunicação serial é feita através de protocolos (INSTRUMENTS, 2018) como:

- *RS-232*: é um padrão para transmissão de dados que define formalmente os sinais de conexão entre um aparelho terminal de dados e um equipamento de comunicação. São definidos sinais de controle, modo de transmissão de dados, entre outros detalhes.

Comparando com os modelos apresentados adiante, esse apresenta uma velocidade de transmissão baixa, entre outras limitações. Porém, por sua simplicidade, o mesmo ainda é utilizado quando a comunicação é ponto a ponto.

- *RS-422*: esse protocolo permite comunicações de até 10 dispositivos. Também permite que as distâncias sejam maiores e são mais resistentes ao ruído devido ao fato que a transmissão diferencial é utilizada.
- *RS-485*: é uma melhoria do protocolo *RS-422*. As mudanças incluem o aumento de 10 para 32 dispositivos e isolamento do ruído externo, que tornaram esse protocolo um dos principais da indústria de automação.

A comunicação *ethernet* é amplamente utilizada nas redes de *internet* comerciais. Ela pode se dar através de pares trançados ou fibra ótica. Com altas velocidades de transmissão e capaz de percorrer grandes distâncias, utiliza protocolos como Modbus/TCP, EtherNet/IP, PROFINET, entre outros (ZHIHONG; PEARSON, 2017). A utilização dessa forma de comunicação permite que algumas facilidades como propagadores de sinal, dois nodos em um link, entre outros, possam ser utilizadas também na indústria.

Por outro lado, dificuldades como a utilização do protocolo TCP³ ou o fato do tamanho do *frame* ser um pouco maior do que o necessário para comunicação entre sistemas de automação se apresentam com esse modelo de comunicação. Essas características podem causar atraso no recebimento de dados pelo sistema, fazendo com que o mesmo não reflita a situação atual do chão de fábrica, causando inconsistências.

Para a análise desenvolvida nesse trabalho, serão utilizados dois sistemas SCADA diferentes, o *BlueWave* e o *Ignition*, que serão descritos mais adiante.

2.2.2 PIMS

PIMS significa *Plant Information Management System* ou Sistema de Gerenciamento de Informações da Planta, em tradução livre. Esse tipo de sistema nasceu na indústria química e petroquímica para armazenar uma grande quantidade de informações por longos períodos de tempo, resolvendo o problema de separação de dados que essas indústrias enfrentavam (OSISOFT, 2016). Os dados são obtidos de diversas fontes e armazenados

³O TCP é um protocolo que provê confiabilidade, comunicação fim-a-fim, entrega ordenada, entre outros serviços. Algumas de suas características como reenvio de pacotes perdidos, apesar de muito úteis, podem gerar atrasos na comunicação em tempo real da qual um sistema SCADA necessita, assim como sua capacidade de se adequar diminuindo a taxa de envio quando a rede apresenta tráfego alto. Mais informações sobre esse protocolo estão disponíveis em (KUROSE; ROSS, 2012)

em um banco de dados de historiador (CONTROLS, 2001). Esse tipo de sistema é capaz de arquivar variáveis analógicas, discretas, strings e grandes objetos binários, tais como fotos, sons, etc.

Sua principal funcionalidade é transformar os grandes volumes de dados acumulados ao longo dos anos de execução, em informações mais facilmente compreendidas por humanos através do processamento e filtragem dos dados obtidos para futura consulta e auxílio nas tomadas de decisão. Esses dados podem ser utilizados, posteriormente, para a replicação de condições anteriores (PRODUCTS, 2014) ou para a resolução de possíveis problemas no chão de fábrica.

Um exemplo de sistema PIMS é o Akeno⁴.

2.2.3 DCS

DCS significa *Distributed Control System* ou Sistema de Controle Distribuído. Esses sistemas foram projetados para automatizar plantas geograficamente distribuídas, onde cada controlador controla uma ou mais máquinas. Esses controladores são capazes de se comunicar entre si e também com a unidade de controle central, que é conectada a todos e aos atuadores. A arquitetura permite que o sistema de uma planta, que é relativamente grande, seja dividido em pequenos subsistemas semi-independentes conectados, que são capazes de continuar funcionando em caso de falha dos outros.

Essa independência se mantém também na comunicação dos sistemas, uma vez que não é necessário que o mesmo protocolo seja utilizado para todos os componentes do sistema.

Exemplos de sistemas DCS são o sistema PROVOX⁵ e o sistema FOX1⁶.

2.2.4 ERP

ERP significa *Enterprise Resource Planning* ou Planejamento de Recursos do Empreendimento, em tradução livre. Esse *software* tem como objetivo auxiliar na administração e interpretação de dados de diversas atividades de negócio. O sistema é dividido em módulos como serviços do consumidor, recursos humanos, produção, distribuição, contabilidade,

⁴Disponível em <https://www.akeneo.com/>

⁵Disponível em <http://www.emerson.com/en-us/catalog/deltav-provox>.

⁶Mais informações disponíveis em <http://s3data.computerhistory.org/brochures/foxboro.fox1.1971.102646169.pdf>.

etc. Tem-se uma visão em tempo real (ou muito próxima disso) de diversos departamentos de uma empresa e, para tornar isso possível, o sistema conta com um banco de dados comum a todos os módulos, visto que o sistema é distribuído.

Uma das principais características do ERP, no âmbito da automação, é seu uso de *staging tables* para a comunicação com o chão de fábrica (SKORKOVSKÝ, 2011). *Staging tables* são, de forma simplificada, tabelas intermediárias no banco de dados. *Staging* é o processo de preparação de dados antes que as informações sejam carregadas em uma tabela oficial (CORPORATION, 2018a). Essas tabelas oficiais são alimentadas e o ERP consome os dados ali inseridos. A consistência entre os bancos de dados dos módulos é algo essencial para o funcionamento correto do sistema.

Exemplos de ERP são os sistemas Epicor ERP⁷ e o Sage Enterprise Management⁸.

2.3 Logs

Um *log*, no contexto da computação, é uma documentação cronológica de eventos relevantes para um sistema. Existem diferentes tipos de logs, como pode ser visto em (CORPORATION, 2018b) e (APACHE, 2018). Os principais tipos são:

- *Log de mensagem*: são arquivos que contêm as mensagens que chegam e saem para consumo quando um sistema cliente-servidor não estiver mais em execução. Esse tipo de mensagem é utilizado para saber quais argumentos foram passados e como os mesmos foram recebidos pelo servidor. Também são úteis para verificar segurança e encriptação, entre outros (CORPORATION, 2017) (CORPORATION, 2018c).
- *Log de transação*: no contexto de banco de dados, *logs* de transação são um histórico das transações realizadas pelo sistema de gerência de banco de dados para que, em caso de falha ou inconsistência, seja possível procurar por ações executadas com erro, podendo tomar ações para recuperar a consistência. (TIPS, 2014)
- *Log de evento*: Esse tipo de *log* é utilizado em um sistema baseado em eventos para ajudar a determinar a causa de um erro, assim como auxiliar na correção do mesmo, identificando o contexto no qual ele ocorreu (JAIN, 1991) (CORPORATION, 2014).

É preciso ter cuidado para não confundir *logs* com rastro de programas. Enquanto

⁷Disponível em <https://www.epicor.com/erp-systems/epicor-erp.aspx>.

⁸Disponível em <http://www.sagex3.com/en>.

os rastros apresentam informações de baixo nível, como exceções, e são usados principalmente por desenvolvedores, logs apresentam informações de alto nível e têm como objetivo examinar o software num nível mais "gerencial". Muitas vezes, *softwares*, como o Splunk (SPLUNK, 2001), são utilizados para auxiliar na leitura dos *logs*.

Existem duas partes no processo de *logging* de um sistema, cada uma com suas dificuldades específicas: a geração e o armazenamento. Quando os *logs* são gerados, é necessário cuidado para que exista um equilíbrio entre o valor do dado a ser salvo e a carga extra que isso adicionará no sistema. Enquanto algumas informações podem auxiliar, por exemplo, na identificação do perfil do usuário em um dado momento, outras podem nunca ser utilizadas. Porém, não há um método exato que permita determinar qual a melhor forma de logar o sistema, já que muito do conhecimento que se tem até hoje sobre esse processo é empírico (FU et al., 2014). O armazenamento das informações logadas também tem seus problemas específicos como, por exemplo, o tempo que determinadas informações ficarão armazenadas e o nível de detalhamento que as mesmas terão.

Logs são usados com diversos objetivos (ORNELAS, 2003), a saber:

- *Debugging*: *logs* são capazes de representar o fluxo de um *software* durante sua execução, auxiliando na tarefa de encontrar *bugs*, pois é possível verificar onde os erros ocorrem e o contexto dos mesmos.
- *Performance*: *logs* também podem ser utilizados para verificar a performance de um sistema, identificando quais recursos são mais utilizados e a carga que o sistema recebe nos diferentes momentos de uso.
- *Segurança*: a partir do histórico de uso de um sistema, pode-se identificar quando foram feitas tentativas de mau uso ou violações no mesmo. Por exemplo, utilizando *logs* pode-se verificar se um único usuário tentou fazer um número "acima do normal" de *logins* em um curto período de tempo, caracterizando um possível ataque de negação de serviço ⁹.
- *Predição*: as informações coletadas podem ser usadas para fazer previsões sobre utilizações futuras do sistema, como uma unidade de processamento mais rápida, ou necessidades do usuário, como aumento de memória de uma determinada parte da infraestrutura.
- *Relatórios e Perfis*: os eventos logados capacitam a criação de perfis muito usados pela área de *business analytics*. Podemos determinar os caminhos percorridos

⁹Ataques de negação de serviço tem como objetivo tornar os recursos de um sistema indisponíveis para os seus utilizadores. Mais informações disponíveis em (INCAPSULA, 2018)

em um determinado software, recursos utilizados, carga de uso e popularidade de determinadas partes do sistema.

As informações coletadas compõem uma grande quantidade de dados que não têm necessariamente a mesma origem. Por exemplo, alguns dados podem vir de um servidor enquanto outros vêm do sistema de armazenamento. Por isso, correlacionar as variáveis pode tornar os *logs* mais valiosos, possibilitando maiores descobertas. Por outro lado, isso pode se tornar um desafio, dados os obstáculos conhecidos apresentados por sistemas distribuídos, como a sincronização de relógios e normalização dos nomes das variáveis capturadas.

Outra característica que pode ser útil em um sistema de *logs* são alarmes. Alarmes podem ser configurados no sistema para que, caso alguma informação coletada esteja fora do esperado ou não seja coletada, os responsáveis por monitorar o mesmo serão avisados para que possam ser tomadas ações para remediar o problema.

Nesse trabalho, nos aprofundaremos nos historiadores, presentes em sistemas de automação e que apresentam as possibilidades de correlação, alarmes e diversas outras facilidades.

2.4 Historiadores

Historiadores são amplamente utilizados integrados a sistemas da indústria de automação. O objetivo de um historiador é possibilitar a consulta e análise de dados de longa data para determinar problemas de performance, automação, controle e equipamentos. Esse software coleta informações provenientes de diferentes sistemas e as armazena em uma base de dados temporal, possibilitando a consulta e centralizando todos os dados em um só local. Ele também permite correlacionar os dados coletados para identificar tendências e relacionamentos, facilitando a análise dos mesmos.

As informações a serem armazenadas são extraídas a partir de pontos de coleta, também chamados de *tags*, que capturam diferentes tipos de dados, além de metadados. Metadados são informações sobre um dado, como qualidade do dado e *timestamp* e são utilizados para determinar se o dado salvo é confiável e quando o mesmo ocorreu (ZENG; QIN, 2016).

A qualidade de um dado é parte importante do sistema. Antes que o dado chegue ao historiador, o SCADA verifica se o mesmo está de acordo com o esperado. Essa

aferição é feita através de diferentes técnicas descritas em (PUROMÄKI, 2010). Uma informação pode ter qualidade ruim por diversos motivos: problemas nos sensores causando medição errônea, interferências na comunicação, entre outros.

A qualidade é tipicamente representada por um número. Por exemplo, se a qualidade for representada em termos de porcentagem, o valor 100 indica que todos os dados do intervalo são bons. Os dados podem ser bons, ruins, incertos, entre outros. Existem outras formas de representação de qualidade, como mostra a Figura 2.2. O *timestamp* utilizado é, geralmente, o *Unix timestamp*, que conta os segundos desde primeiro de janeiro de 1970.

Figura 2.2: Fonte: (SYSTEMS, 2005). Exemplo de uma tabela representando a qualidade da informação em um historiador

Hex	Dec	Name	Description
0x00	0	Good	Good value.
0x01	1	Bad	Value was marked as invalid.
0x10	16	Doubtful	Value is uncertain.
0x85	133	Initial Value (Good)	Initial value for a delta request.

O historiador BW, foco desse trabalho, permite que sejam definidas diferentes tipos de coleta:

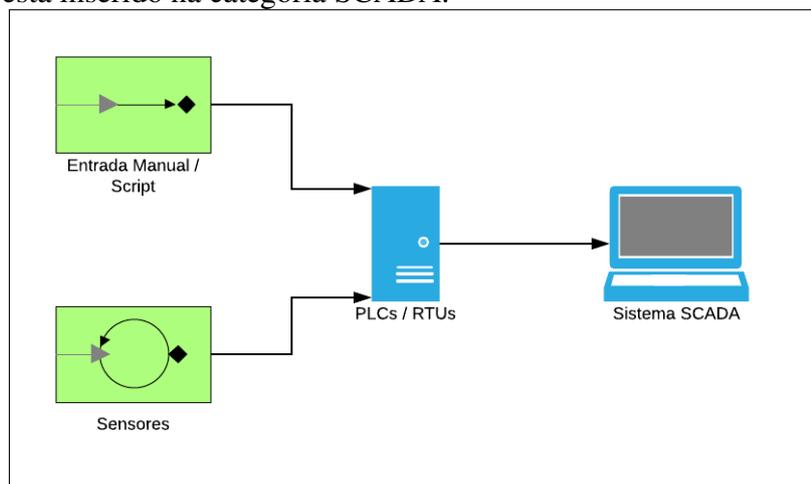
- *Coleta em frequências fixas*: quando é definido um intervalo de tempo regular para a coleta. Por exemplo, a cada segundo, será coletado um valor.
- *Coleta baseada na mudança de um valor*: quando a variável só é coletada caso ocorra uma mudança em seu valor. Por exemplo, se o valor em observação estiver em '1', quando ele for pra '0', a mudança será gravada.
- *"Gatilho" de coleta*: quando é definido um intervalo de valores aceitáveis para uma *tag* e esse valor é extrapolado. Por exemplo, se o valor que está sendo observado estiver entre '1' e '27', o mesmo será coletado e salvo no banco de dados.

Um historiador coleta uma grande quantidade de dados e os mesmos devem ser mantidos por longos períodos de tempo. Logo, é necessário que a forma de armazenamento e a compressão dos dados sejam aprimoradas de maneira que seja possível consultá-los de forma eficiente, quando necessário, sem utilizar muito espaço no disco. Dessa forma, é possível manter os dados por mais tempo sem a necessidade de um aumento no armazenamento.

Os dados são coletados através de uma rede de aquisição representada na Figura 2.3. Essa rede se inicia nos sensores conectados aos objetos historiados. Os sensores percebem alterações nos objetos e transmitem dados, através de um protocolo de comunicação,

aos controladores. O controlador, que pode ser representado por uma RTU ou um PLC, tem duas tarefas: comunicação com o historiador e execução das rotinas de controle. Por exemplo, ao receber os dados, o historiador verifica em quais tabelas a informação capturada está presente e se a mesma será guardada de acordo com os critérios de coleta mostrados acima. Se for feito o salvamento, os dados presentes em todas as *tags* que compõem aquela tabela serão salvos também, tornando possível a análise de contexto do momento.

Figura 2.3: Fonte: autora. Representação de uma rede de aquisição. Nessa figura, o historiador está inserido na categoria SCADA.



Sensores são definidos como "elementos de um sistema de medição que são diretamente afetados por um fenômeno, corpo ou substância que contém a grandeza a ser medida" (TERGOLINA, 2015). Por exemplo, a pressão no interior de uma das máquinas do chão de fábrica é um fenômeno que afeta um sensor.

Uma rede de aquisição também é capaz de responder a certos estímulos através de rotinas de controle. Por exemplo, quando uma caldeira passa de sua temperatura máxima segura, o sistema pode resfriá-la antes mesmo de enviar tais informações de temperatura ao historiador. Atuadores são componentes elétricos, mecânicos, pneumáticos ou hidráulicos capazes de, a partir de sinais elétricos, atuarem nos sistemas nos quais estão instalados. Um exemplo disso é o motor elétrico, que converte sinais elétricos em rotação, provendo mais torque ou velocidade, colocando alguma parte do sistema em movimento (OGATA, 2001).

Historiadores lidam com duas definições diferentes de tabela. A primeira, citada no texto acima, é a ideia de uma tabela criada pelo usuário dentro do sistema do historiador. Ela tem como objetivo correlacionar variáveis presentes no sistema. Por exemplo, se um grupo de caldeiras é interdependente, podemos correlacionar as informações sobre as

mesmas em uma tabela de caldeiras. O segundo conceito é o de tabelas de um banco de dados, as quais serão usadas para salvar as informações recolhidas. Essas tabelas serão criadas a partir das tabelas definidas pelo engenheiro de projeto no sistema. Para fins de terminologia, as tabelas criadas pelo usuário no sistema serão definidas como Tabela do Historiador.

2.5 BlueWave

O *BlueWave* é um sistema SCADA, desenvolvido pela Altus, empresa da indústria de automação que tem parceria com a UFRGS através do projeto SDCD. Os sistemas SCADA integram diferentes funcionalidades fundamentais para o chão de fábrica. Eles são compostos por diversos controladores autônomos que são distribuídos pelo sistema e se comunicam com o chão de fábrica e os historiadores que são parte do SCADA.

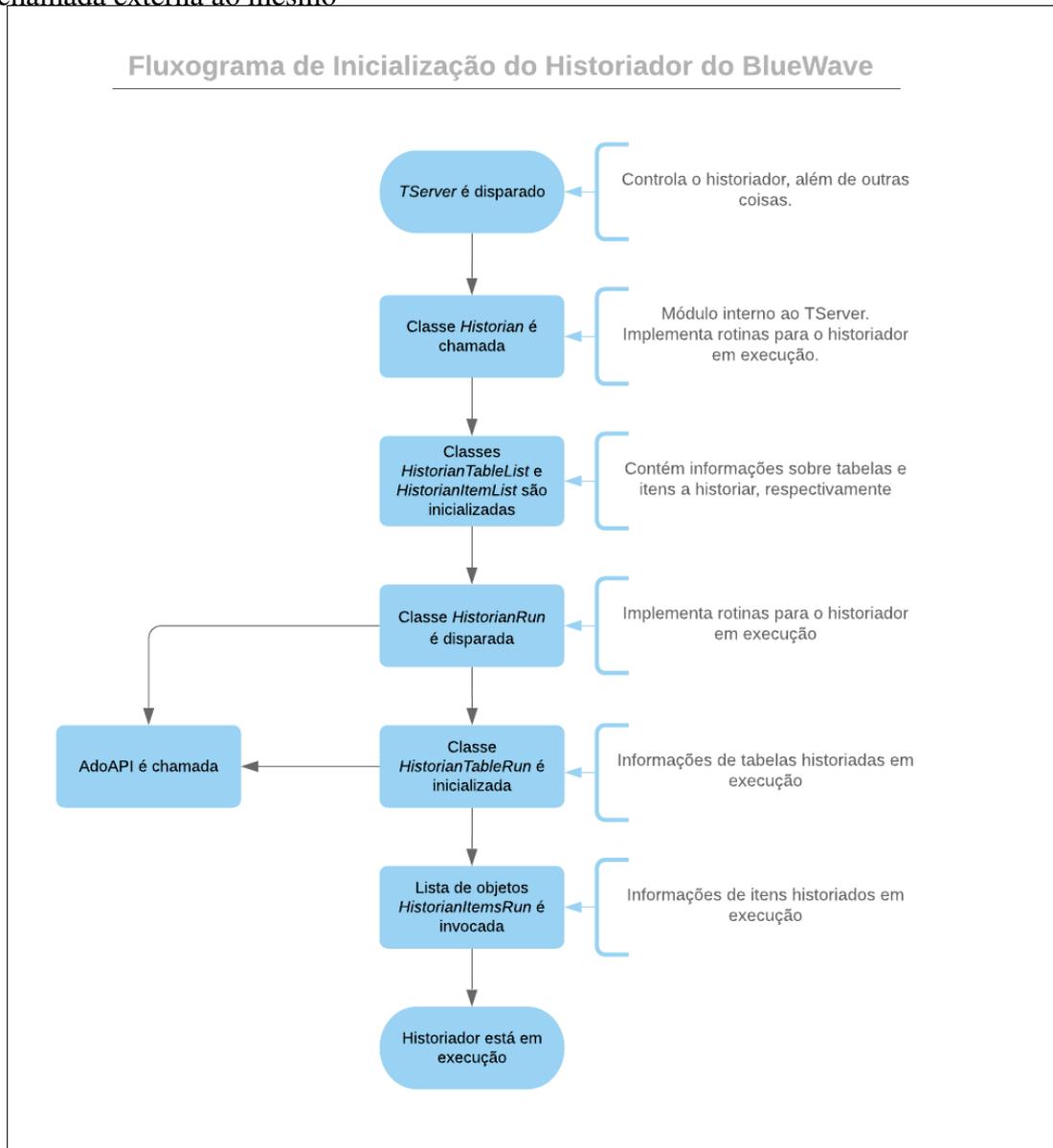
O *BlueWave*, em particular, utiliza dois protocolos de comunicação:

- *Modbus*: "O Modbus é um protocolo de comunicação desenvolvido pelos sistemas Modicon. Em termos simples, é um método usado para transmitir informações através de linhas seriais entre dispositivos eletrônicos. O dispositivo que solicita a informação é chamado de Modbus Master e os dispositivos que fornecem informações são Modbus Slaves."(ORGANIZATION, 2018). Esse é o mais rápido, pois faz requisições o tempo todo para verificar se novas informações estão disponíveis. Esse tipo de comunicação retorna os dados sem nenhum tipo de processamento ao historiador.
- *OPC DA*: "OPC DA significa *Open Platform Communications Data Access* ou Plataforma de Comunicação Aberta Dados de Acesso. Essa é uma especificação do OPC Foundation que define como os dados em tempo real podem ser transferidos entre uma fonte de dados e um *data sink* (por exemplo: um PLC e um SCADA) sem que nenhum deles tenha que conhecer o protocolo nativo um do outro"(FOUNDATION, 2018). Esse protocolo de comunicação trabalha com dados em tempo real, sendo capaz de extrair seu valor, o *timestamp* e a qualidade dessa informação, além de não criar conflitos quando faz a requisição.

O historiador do *BlueWave* é o ponto principal deste trabalho e, por isso, será descrito em maior detalhe a seguir. Dentro do historiador, temos dois tipos diferentes de objetos:

- *HistorianItem*: representa um objeto a ser historiado.
- *HistorianTable*: representa as tabelas geradas pelo criador do projeto que contém informações sobre quais *tags* serão processadas e armazenadas.

Figura 2.4: Fonte: Documentação dos desenvolvedores do BlueWave. Fluxograma representando como as classes são chamadas dentro do historiador, começando pela chamada externa ao mesmo

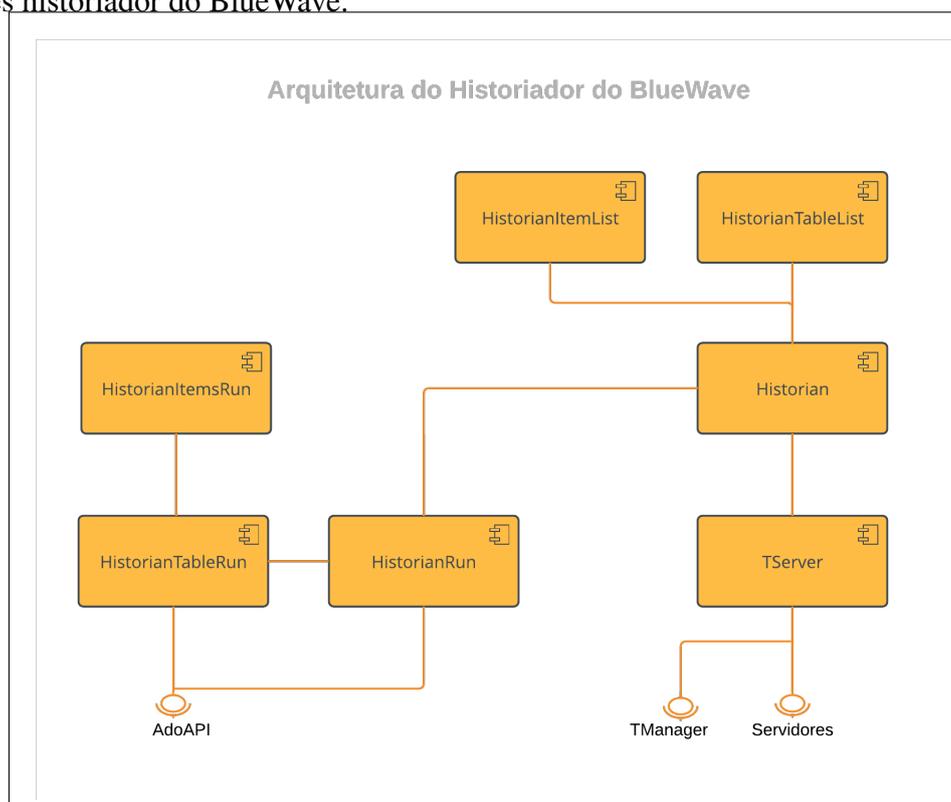


O fluxograma da Figura 2.4 representa o fluxograma de inicialização do historiador dentro do *BlueWave*. Enquanto algumas classes têm poucas responsabilidades dentro do sistema, outras são mais complexas. A classe *HistorianRun* possui as funções necessárias para controlar o historiador durante suas execuções. Ela é responsável pela preparação de

todas as rotinas SQL¹⁰ que serão executadas, controle do arquivo *.tHistorian*¹¹, gerenciamento, paralelização das tarefas, sincronização de redundância (quando a mesma existe) e a execução de uma tarefa chamada *SaveAndForwardToDatabase*. Essa chamada trata de casos em que se executa um projeto localmente e deseja, de tempos em tempos, fazer a fusão das alterações locais com o banco de dados do servidor em si. *HistorianTableRun* faz a amostragem das informações e salva as mesmas. Ela é composta por funções de salvamento e compressão de dados. Por último, *AdoAPI* é responsável por resolver a maior parte da preparação e execução de comandos de banco de dados no BlueWave. Ela prepara e executa *queries* com diferentes formatos de acordo com o necessário para cada provedor. Essa classe é utilizada por diferentes partes do sistema, não só o historiador.

O historiador do BW é estruturado, em questão de classes, conforme mostrado na Figura 2.5.

Figura 2.5: Fonte: Documentação dos desenvolvedores do BlueWave. Diagrama de classes historiador do BlueWave.



A integração do sistema com o historiador é feita pelas classes TManager e TServer, como visto na Figura 2.6. O módulo TManager é responsável por edição de

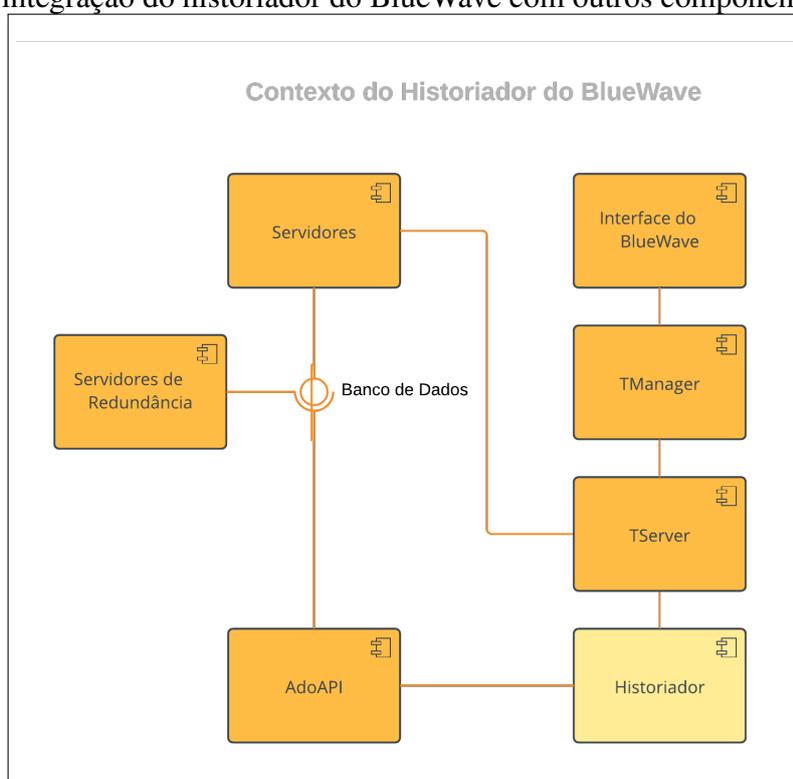
¹⁰SQL significa "Structured Query Language", ou linguagem de consulta estruturada. Essa é a linguagem mais utilizada para sistemas de gerenciamento de banco de dados executados em minicomputadores e mainframes (BEAL,).

¹¹O arquivo *.tHistorian* armazena as informações geradas em tempo de execução.

projetos, criação de *tags*, telas de operação, *scripts*¹², canais de comunicação, configuração de alarmes e bases de dados, configuração do historiador, etc.. Também dentro desse processo é que ocorre a implementação da redundância do servidor. O TServer controla os projetos em execução. Ele pode ser executado em um contexto local¹³, em um contexto de Cliente-Servidor¹⁴ ou em um contexto de servidor de redundância¹⁵. Ele controla as rotinas críticas para a execução do projeto, entre elas, o próprio historiador.

Na Figura 2.6, pode-se ver como o historiador se conecta com o resto do sistema.

Figura 2.6: Fonte: Documentação dos desenvolvedores do BlueWave. Diagrama de módulos da integração do historiador do BlueWave com outros componentes do sistema.



2.6 Ignition

Ignition é o software modular da *Inductive Automation* que possibilita que diferentes funcionalidades sejam acopladas para que o sistema atenda as necessidades do usuário (AUTOMATION, 2018b). Essa modularidade tem dois níveis, a modularidade arquitetural

¹²Os *scripts* possibilitam ao engenheiro a edição do código responsável pelos pontos de coleta, criação de dados fictícios, etc.

¹³O contexto local é utilizado para testar o cliente sem afetar o servidor

¹⁴O cliente se conecta com estações de manutenção, operação e engenharia, sendo capaz de enviar informações para diferentes partes da planta

¹⁵No contexto de redundância, existe uma instância do módulo em cada servidor realizando sincronização de dados

e a modularidade de funcionalidades. Esse sistema é independente de sistema operacional, tem todas as funcionalidades centrais que caracterizam um SCADA dentro de uma plataforma central do sistema e o *deploy* dos projetos é feito a partir de uma plataforma web ao invés de utilizar um sistema instalado na máquina, o que possibilita o uso do mesmo projeto em diversas máquinas.

Dentro dessa modularidade arquitetural, existem três camadas (AUTOMATION, 2018c):

- *Módulo de Plataforma*: esse módulo traz as funcionalidades de comunicação (como, por exemplo, protocolo OPC), licença, sistema de *logs*, autenticação, entre outras funções básicas utilizadas pelo sistema SCADA para execução de suas tarefas, como conexão com a base de dados.
- *Módulo de HMI/SCADA*: essa camada traz o SCADA em si. Suas telas de comunicação com os operadores e engenheiros. Cada módulo de funcionalidade adicionado é capaz de estender o núcleo do sistema presente como, por exemplo, a capacidade de acessar o *software* através de um celular.
- *Módulo de Sistema de Execução de Manufatura*: aqui estão presentes módulos de indústrias mais específicas como, por exemplo, a indústria elétrica. Eles possuem uma licença separada dos outros dois. Também é possível que os engenheiros criem seus próprios módulos.

A *Inductive Automation* tem uma página em seu *web site* dedicada à venda de módulos de extensão do sistema com as categorias: alarmes, análise de dados, realidade aumentada, gráficos, conectividade, mapeamento, monitoramento, ferramenta de escritório, criação de *scripts* e simulação. Os diferentes módulos disponíveis para cada subcategoria estão disponíveis em <https://inductiveautomation.com/moduleshowcase/>.

Para esse estudo, o *software* foi utilizado sem adição de módulos de Sistema de Execução de Manufatura, ou seja, apenas o sistema SCADA sem extensões.

O *Ignition* tem duas formas de lidar com dados históricos (LLC, 2018):

- *Tag Historian*: *Tags* podem ser configuradas para salvar informações.
- *Transaction groups*: Grupos de itens registrados em um ciclo de execução.

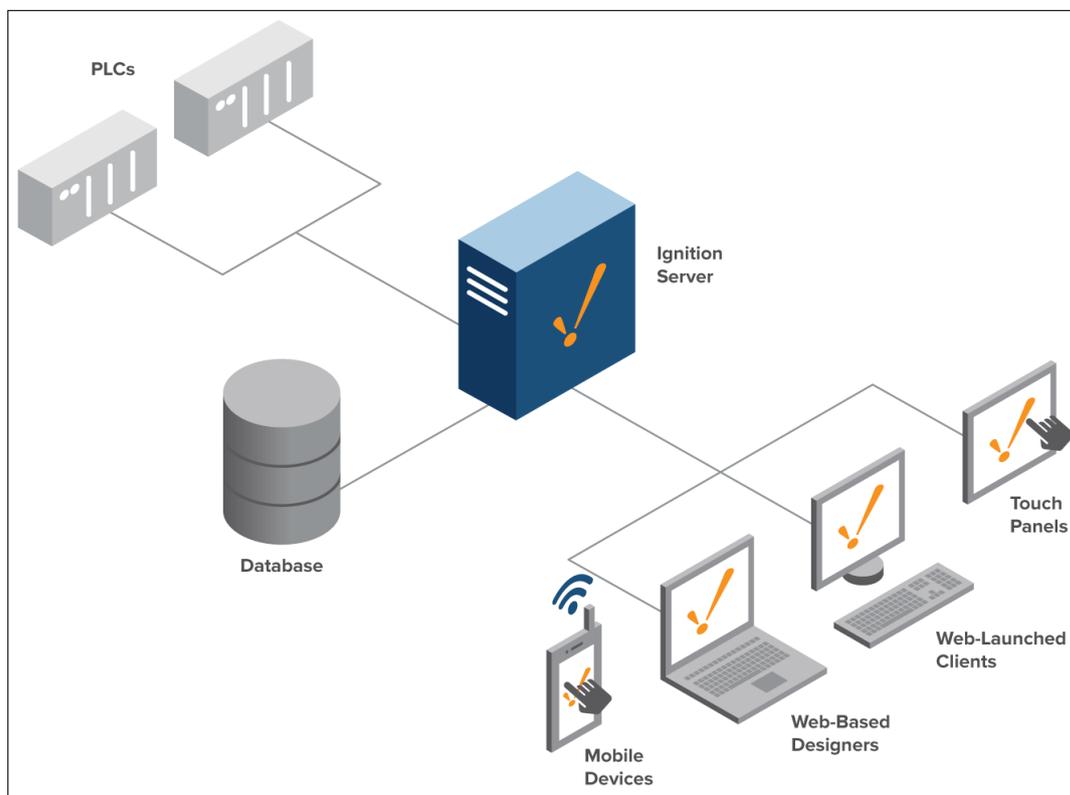
Para o estudo desenvolvido nesse trabalho, será utilizado o modo *Tag Historian*.

Esse sistema é baseado em servidores, o que possibilita que um computador tenha o SCADA instalado e outros possam acessá-lo. O grande diferencial do *Ignition* é a possibilidade de adicionar e desenvolver módulos que estendem sua funcionalidade para

realizar as atividades pretendidas pelo cliente. Essa arquitetura traz muitas possibilidades, como uma separação entre os servidores que fazem a conexão com os computadores nos quais os técnicos de projeto e manutenção trabalham para monitorar o sistema e os sistemas SCADA de fato.

A arquitetura do *Ignition* deste estudo está representada na Figura 2.7. Essa representação não está mostrando a modularidade do mesmo.

Figura 2.7: Fonte: (AUTOMATION, 2018c). Imagem ilustrativa da arquitetura do *Ignition* utilizada nesse trabalho.



Para o desenvolvimento desse trabalho, serão seguidos os passos definidos na Seção 2.1 e os conhecimentos presentes nesse capítulo para a formulação, execução e análise do teste de desempenho dos historiadores dos dois sistemas SCADA descrito.

3 DESENVOLVIMENTO

A partir dos passos apresentados na Seção 2.1, para avaliar o desempenho o historiador de um sistema SCADA é necessário, primeiramente, definir quais partes do sistema serão testadas. Para o escopo desse trabalho, o alvo dos testes serão os historiadores dos sistemas *BlueWave* e *Ignition*. O primeiro passo deste trabalho será avaliar se o BW realmente utiliza mais espaço no banco de dados e, se isso for verdade, determinar a causa dessa ocorrência.

À medida em que os dados vão sendo inseridos na tabela, o sistema os aloca no banco de dados. Os clientes do *BlueWave* indicaram para a empresa que o desenvolve, a Altus, que, comparando com outros sistemas SCADA disponíveis na indústria, o mesmo utiliza grande espaço em disco.

O uso de espaço em disco é um tópico bem discutido pelas empresas que desenvolvem sistemas que envolvem historiadores (AUTOMAÇÃO, 2012a) dado o seu objetivo de armazenar uma grande quantidade de dados por um longo período de tempo. As tabelas são formatadas de maneira que haja um balanceamento entre a quantidade de dados armazenada e sua velocidade de busca, outro gargalo conhecido nesse tipo sistema. Para esse trabalho, será analisado apenas o uso de disco e a forma como as tabelas são construídas no BD.

3.1 Definição dos objetivos do estudo e os limites do sistema.

O objetivo do estudo deste trabalho é determinar se o uso de espaço em disco do BW está acima de um de seus competidores, o *Ignition*.

Para que seja possível testar esse sistema, é necessário compreender como as classes que o compõe se relacionam e os diferentes caminhos para que cada uma delas seja acessada. Como mostrado na Figura 2.5 e descrito na seção 2.5, é possível determinar o fluxo de informações dentro do *software*. Essas informações, apesar de não serem pertinentes para este estudo, são importantes para o trabalho futuro de, se houver um uso exacerbado de espaço, apontar possíveis causas.

No escopo desse trabalho, os sistemas SCADA não serão testados como um todo, apenas seus historiadores. Mais especificamente, como as tabelas inseridas pelos usuários são criadas dentro do banco de dados e, então, como o espaço em disco é utilizado.

3.2 Definição dos serviços do sistema e os possíveis resultados.

Inicialmente, é feita a configuração da conexão entre o banco de dados e o sistema SCADA através dos passos contidos nos manuais dos mesmos. Para o escopo desse trabalho, apenas dois serviços serão avaliados: o de criação de tabelas a partir do *input* do engenheiro e o de inserção de dados coletados no banco de dados.

Para atingir o objetivo definido em 3.1 e testar os serviços supracitados, foram levantadas duas hipóteses:

- *Hipótese 1 - A estrutura de tabela criada pelo BlueWave não está em uma forma ótima¹: o BlueWave cria automaticamente as tabelas do banco de dados a partir das tabelas do historiador, criadas pelo responsável pelo projeto.*
- *Hipótese 2 - O BlueWave está armazenando os dados de forma que muito espaço desnecessário é utilizado:* Como visto em 2.4, é necessário que a forma que os dados são salvos seja otimizada para que utilize a menor quantidade de armazenamento possível. Para isso, o espaço utilizado pelo BW será comparado com o do Ignition para que seja possível avaliar se os valores são condizentes.

A partir da análise dos resultados que testem essas duas hipóteses, será possível determinar:

- Se o *BlueWave* utiliza mais espaço em disco que o *Ignition* em quais cenários isso acontece.
- Possíveis anomalias na rede de aquisição e armazenamento de dados que torne o uso de espaço em disco ineficiente.

3.3 Seleção de métricas de desempenho.

A métrica de desempenho selecionada foi o espaço em disco utilizado pelas tabelas medido em Kilobits. Essa métrica é obtida através da geração de um relatório diretamente na ferramenta de gerenciamento de banco de dados, o *Microsoft SQL Server 2014*.

¹ nesse caso, a forma ótima seria a que permite a atualização de todos os pontos de coleta sem desperdício de espaço.

3.4 Definição dos parâmetros do sistema e do *workload*.

Para que um teste de desempenho que utiliza medição seja válido, é preciso que ele represente o mais próximo possível a situação de uso real do sistema. Um historiador tem diferentes situações de estresse possíveis, tais como:

- *Valores fora de ordem*: os dados recebidos por um SCADA são enviados a um banco de dados, que sofre compressão de acordo com intervalos de tempo pré-determinados. Se um valor for recebido depois que a janela de tempo ao qual o mesmo pertence já sofreu essa compressão, o sistema precisa descomprimir, inserir o valor na posição correta e recomprimir, o que pode ser custoso.
- *Avalanche de dados*: essa situação acontece quando o historiador do sistema recebe muito mais dados do que o esperado. Por exemplo, se um sistema é responsável por 350 caldeiras e todas elas têm valores acima do limiar determinado para temperatura, que é coletada por *tags* que utilizam o gatilho como forma de coleta, o sistema é responsável por coletar as temperaturas de cada uma delas, criando um fluxo muito grande. Além das situações citadas acima, um historiador real atende diferentes tipos de variáveis (inteira, binária, string, etc.) com picos de demanda, quando chegam muitas variáveis.
- *Avalanche de tags*: A criação de *tags* não se limita ao momento em que o sistema é instalado e as mesmas podem ser criadas enquanto o mesmo está sendo executado. Se um grande número de pontos de coleta é criado dessa forma, isso pode causar uma extenuação do sistema.

Outro possível gargalo de um historiador é seu banco de dados. Espera-se que as informações coletadas sejam mantidas por um longo período de tempo, podendo chegar a anos. Logo, é necessário que a estrutura das tabelas de banco de dados seja projetada de forma que não apenas comporte os dados, mas também busque uma forma de compressão para que não seja ocupado muito espaço em disco.

Se os arquitetos do sistema tiverem feito a opção de realizar a compressão de dados, a cada busca, os dados devem passar por um processo de descompressão, busca no banco e, após as operações, devem ser comprimidos novamente.

Existem muitos possíveis gargalos para o historiador de um sistema SCADA. Como explicitado na seção 2.1, é preciso definir um foco para que os testes possam avaliar corretamente a reação do sistema frente um certo cenário.

Para esse trabalho, as inserções serão feitas em intervalos regulares e sempre com valores dentro do intervalo esperado. Essa decisão foi tomada devido a necessidade de evitar perdas na análise de como, em condições ideais, é utilizado o disco.

A avaliação será feita utilizando a configuração que, a cada 30 segundos, verifica o estado atual do sistema e insere o valor dos pontos de coleta no banco de dados. Para que seja possível avaliar o seu comportamento, foi criado um *script* que, nesse mesmo intervalo de tempo, modifica o valor das variáveis.

3.5 Seleção dos fatores e seus valores.

Esses cenários foram criados baseadas nas hipóteses definidas na Seção 3.2:

- *Hipótese 1 - Avaliação das tabelas de banco de dados criadas:* essa avaliação tem como objetivo a análise das tabelas criadas pelos diferentes historiadores através da comparação da estrutura determinada pelos dois sistemas. Essa avaliação pode nos mostrar que o formato escolhido traz efeitos negativos ao uso do espaço no banco a partir da análise dos melhores e piores casos.
- *Hipótese 2 - Avaliação do espaço utilizado através de inserção de dados no banco:* após cada um dos testes, será feita uma verificação de espaço utilizado no disco e no banco para determinar se as formas de preenchimento das tabelas têm impacto nas diferenças observadas pelos clientes. Esse cenário permitirá verificar se o espaço usado pelas tabelas do BW é maior, menor ou igual ao utilizado pelas tabelas do *Ignition*.

Para essas análises, serão criadas, em cada sistema, 500 pontos de coleta² de números inteiros que terão seu valor modificado durante as avaliações. Nenhum dos dados gerados será fora do intervalo esperado, já que não é objetivo desse teste ver como o sistema se comporta em situações adversas.

3.6 Seleção técnicas de avaliação.

A análise proposta nesse trabalho utiliza um *benchmark* de indústria, comparando o BlueWave com o Ignition e, para esse trabalho, serão feitas duas avaliações que podem

²O número de pontos de coleta foi determinado pela limitação no número de colunas no Microsoft SQL Server.

apresentar diferentes deficiências dos sistemas.

3.6.1 Primeira Hipótese

Para verificar a primeira hipótese feita em 3.5, são analisadas as tabelas criadas pelos sistemas SCADA. Essa verificação tem por objetivo uma avaliação de como as tabelas são formatadas e a definição de seus piores e melhores cenários.

3.6.2 Segunda Hipótese

Dada a segunda hipótese feita, foram criados três cenários para avaliar sua veracidade:

- *Mudança de valor de um ponto da tabela*: essa avaliação tem como objetivo determinar o quão eficientes os sistemas são quando apenas um ponto de coleta tem seu valor modificado e precisa ser atualizado nos registros.
- *Mudança de valor de todos os pontos de coleta da tabela uma vez*: esse teste será utilizado para verificar se há alguma anormalidade³ quando são inseridos grandes volumes de dados apenas uma vez.
- *Mudança de valor de todos os pontos de coleta da tabela várias vezes*: nesse cenário será possível verificar se, durante a execução contínua, os *softwares* se comportam de maneira esperada, ou seja, a cada mudança de valores, atualizam seus registros. Também será possível verificar em um cenário mais próximo do real como o uso do espaço é feito

3.7 Seleção do *workload*

Conforme os cenário de uso descritos na Seção 3.6, as seguintes cargas foram definidas para os experimentos realizados para esse estudo.

3.7.1 Primeira Hipótese

Nessa hipótese, não haverá *workload* pois se trata de uma análise estática.

³Uma anormalidade pode ser perda de dados, duplicação de dados, entre outros.

3.7.2 Segunda Hipótese

No primeiro cenário citado na Seção 3.6, apenas um ponto de coleta terá seu valor modificado para outro dentro do intervalo esperado pelo sistema.

O segundo cenário terá o *workload* composto apenas por um inteiro dentro do intervalo esperado que será que aplicado a todos os pontos de coleta.

Para o teste do terceiro cenário, serão feitas três rodadas de teste de trinta minutos cada⁴ nas quais a cada trinta segundos os valores serão modificados através de um *script* que garantirá que a quantidade de dados gerada ao final do tempo dos testes será a mesma. Ao final de cada uma dessas rodadas, serão coletados relatório de uso de espaço em disco (em Kb) geradas pelo próprio sistema de gerenciamento de banco de dados nos quais as tabelas estão sendo armazenadas.

3.8 Projeção dos experimentos.

Com o intuito de manter o ambiente o mais semelhante possível, os dois sistemas SCADA serão conectados à um banco de dados *Microsoft SQL Server 2014* e os testes realizados no mesmo computador. O computador utilizado tem as seguintes especificações técnicas:

- Processador: Intel(R) Core(TM) i7-6500U CPU @ 2.50 GHz
- Memória RAM: 8,00GB (utilizável: 7,87 GB)
- Tipo de Sistema: Sistema operacional de 64 bits, processador com base em x64
- Espaço em Disco: 1TB (utilizável: 500 GB)
- Navegador⁵: Google Chrome

3.8.1 Primeira Hipótese

Para a realização deste experimento, será necessário seguir o manual de instalação do BW (AUTOMAÇÃO, 2012b) e do *Ignition* (AUTOMATION, 2018a) para que as tabelas sejam criadas no banco de dados e possam ser analisadas.

⁴O tempo de cada rodada foi definido como meia hora devido às limitações de uso da versão *trial* do *Ignition*

⁵É necessário especificar o navegador pois o *Ignition* é uma aplicação Web e o *Chrome* foi usado para acessar o servidor que, no caso, era um *localhost*

3.8.2 Segunda Hipótese

- *Mudança de valor de um ponto da tabela*: neste caso, um ponto de coleta terá seu valor modificado manualmente afim de que seja possível verificar o efeito disso no espaço usado.
- *Mudança de valor de todos os pontos de coleta da tabela uma vez e mudança de valor de todos os pontos de coleta da tabela várias vezes*: para estes dois cenários, foi criado um *script* dentro dos historiadores⁶ com uma variável indicando o intervalo de tempo no qual os valores devem ser modificados.
 - Para o primeiro experimento supracitado, o sistema será inicializado, os valores modificados e o sistema parado, sendo feita a geração do relatório na ferramenta de gerenciamento de BD.
 - Para o último, foi colocado um limite de 60 iterações⁷.

3.9 Análise e interpretação dos dados.

3.9.1 Primeira Hipótese

A tabela do *Ignition* na qual são inseridos os registros com os valores das *tags* tem os campos, mostrados na Tabela 3.1:

Tabela 3.1: Tabela representando a estrutura de dados criada pelo *Ignition*

Nome do Campo	Informação do Campo
tagid	Identificação do ponto de coleta.
intValue	Valor inteiro coletado para aquela tag.
floatValue	Valor de ponto flutuante coletado para aquela tag.
stringValue	Valor de <i>string</i> coletado para aquele ponto de coleta.
dateValue	Valor de data coletado para aquela tag.
dataIntegrity	Representa a qualidade do dado coletado.
t_stamp	<i>Timestamp</i> no qual o dado foi coletado.

⁶Os dois sistemas SCADA escolhidos permitem que o usuário desenvolva códigos dentro da aplicação, permitindo a criação de dados de entrada, alarmes, entre outros.

⁷Cada rodada dura 30 minutos e as mudanças são feitas a cada 30 segundos, logo, serão 60 modificações.

A modelagem escolhida pelo *Ignition* permite que, quando um ponto de coleta é alterado, apenas uma nova linha seja adicionada à tabela. Porém, quando todas as *tags* que compõem a tabela têm o valor alterado, essa quantidade de linhas precisa ser adicionada para que o sistema represente a realidade do chão de fábrica.

O *BlueWave* tem uma outra abordagem ao armazenamento dos dados. A tabela criada pelo *BlueWave* é composta pelas colunas mostradas na Tabela 3.2:

Tabela 3.2: Tabela representando a estrutura de dados criada pelo *BlueWave*

Nome do Campo	Informação do Campo
id	Identificação do ponto de coleta.
UTCTimestamp_Ticks	Timestamp no qual o dado foi coletado.
LogType	Representa como a coleta é feita.
NotSync	Representa se a variável está sincronizada com outras.
NodeX	Salva o valor assumido pela variável em um determinado momento.
NodeX_Q	Apresenta a qualidade do dado coletado.

A variável *LogType* pode assumir três valores inteiros diferentes, cada uma representando um dos três tipos de coleta disponíveis: no começo da execução, durante a execução ou no fim da execução.

Essa representação de dados exige que, a cada atualização de uma variável que participa dessa tabela, todas devem ter seu valor salvo, mesmo que seus valores não tenham sofrido nenhuma alteração.

A partir de uma análise estática das tabelas criadas pelos dois sistemas para armazenar os registros, é possível determinar o pior caso e o melhor caso de uso em espaço de disco para cada uma delas. Quanto ao *Ignition*, o melhor caso é quando apenas um ponto de coleta tem seu valor alterado e o pior ocorre quando todas as *tags* tem seu valor alterado. Para o *BlueWave*, o melhor caso é quando todas as *tags* têm seu valor atualizado, pois é possível registrar todas essas mudanças em uma linha, e o pior caso acontece se apenas um ponto de coleta tem seu valor alterado, pois é preciso salvar novamente todos os valores presentes na tabela.

Não é possível determinar se um sistema tem um formato de armazenamento melhor que o outro através de uma análise estática, apenas determinar os piores e melhores casos para que seja possível utilizá-los para testes.

3.9.2 Segunda Hipótese

3.9.2.1 Atualização de um valor da tabela

As tabelas de relatório geradas pelos *Microsoft SQL Manager 2014* estão disponíveis para consulta no Apêndice A.

É possível ver que o *BlueWave* utiliza 16Kb para atualizar o valor de um ponto de coleta, enquanto o *Ignition* usa menos de 1Kb, pois não é feita alocação de espaço para as informações inseridas. Isso ocorre pela diferença na formação das tabelas dos sistemas. Enquanto o primeiro necessita uma atualização de todas as 1004 colunas, mesmo que a maioria das informações adicionadas sejam cópia de dados anteriores, o segundo adiciona apenas uma linha com sete colunas.

Como foi definido anteriormente, esse é o pior caso para o historiador do *BlueWave*.

3.9.2.2 Atualização de todos os valores da tabela uma vez

As tabelas de relatório geradas pelos *Microsoft SQL Manager 2014* estão disponíveis para consulta no Apêndice B.

Para esse cenário, foi feito um *script* que atualizava o valor de todos os 500 pontos de coleta presentes em uma tabela do banco de dados do historiador. Analisando os valores do relatório de uso de disco das tabelas do banco de dados, é possível determinar que o *BlueWave* utilizou 32Kb para historiar os 500 pontos de coleta e o *Ignition* usou 32Kb.

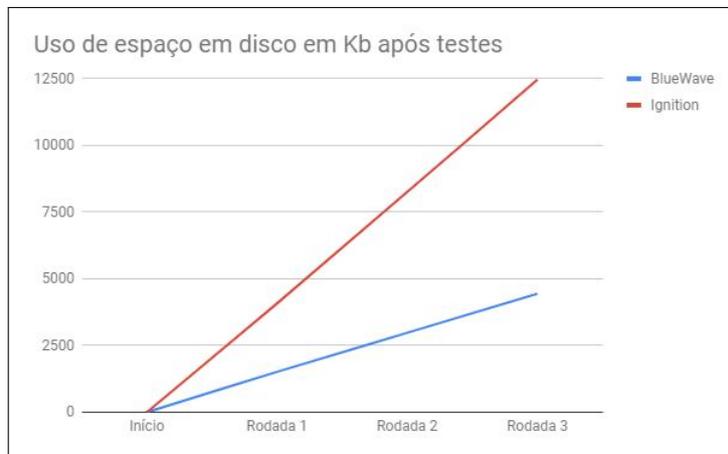
Esse teste mostra que, no pior caso do *Ignition*, a sua utilização de disco foi a mesma do *BlueWave* em seu melhor caso.

3.9.2.3 Atualização de todos os valores da tabela diversas vezes

A partir dos testes de longa duração feitos nos historiadores dos dois sistemas SCADA, foi criada uma tabela para analisar a diferença do uso de espaço em disco, considerando que os bancos estivessem vazios no momento de início dos testes.

Não foi possível realizar os testes com as tabelas vazias porque é preciso confirmar que a conexão com o sistema SCADA está correta e as tabelas estão tendo um comportamento esperado como, por exemplo, que cada modificação é registrada no banco de dados. As tabelas de relatório geradas pelos *Microsoft SQL Manager 2014* estão disponíveis para consulta no Apêndice C.

Figura 3.1: Relatório de uso de espaço em disco (em Kb) após os testes de múltiplas inserções



Utilizando a Figura 3.1⁸ é possível verificar que, dado o teste de melhor caso do *BlueWave*, o mesmo utiliza menos espaço que o *Ignition* quando uma grande quantidade de *tags* são modificadas por um período de tempo. O teste de inserção de um único valor demonstra que os dois sistemas utilizam o mesmo espaço em disco para inserção, logo, é possível concluir que o *BlueWave*, quando está operando em condições normais, utiliza menos disco para realizar as mesmas operações que o *Ignition*. Porém, este teste identificou um outro problema que será detalhado na sequência.

3.10 Apresentação dos resultados.

Dadas as análises apresentadas na Seção 3.9, pode-se concluir que as hipóteses foram refutadas e nem a estrutura das tabelas do BW nem o seu desempenho nos testes apontaram um problema, visto que seu uso de espaço em disco foi inferior ao uso do *Ignition* durante os testes de longa duração. Pode-se afirmar que os resultados foram satisfatórios e não será necessária outra iteração de testes. Porém, um problema foi encontrado no *BlueWave*.

Durante a preparação do ambiente de teste, foi feita a configuração do banco de dados e sua conexão com o *BlueWave*, foi possível notar que uma das tabelas não estava se comportando da forma esperada. A partir da análise do relatório do uso de espaço em disco pelas tabelas do banco de dados, foi possível verificar que a tabela *dbo.DBtesting3* estava com uma alocação de memória muito superior quando comparada com as outras. Considerando que todas as tabelas foram criadas a partir do mesmo script,

⁸Números absolutos estão disponíveis em uma tabela no Apêndice D

utilizando arquivos iguais e terem sofrido o mesmo número de atualizações, a causa dessa divergência foi investigada.

Figura 3.2: Relatório das tabelas do BlueWave durante a preparação do banco de dados

Nome da Tabela	Nº de Registros	Reservado (KB)	Dados (KB)	Índices (KB)	Não Usado (KB)
dbo.DBTesting1	7	104	56	48	0
dbo.DBTesting2	7	104	56	48	0
dbo.DBTesting3	261	2.152	2.088	48	16
dbo.DBTesting4	7	104	56	48	0
dbo.Table1	0	0	0	0	0

Ao investigar a tabela no banco de dados, foi possível verificar a causa do maior uso de disco por esta tabela. Sem qualquer gatilho diferente de coleta, as tabelas apresentam um grande número de coletas com *timestamp* diferente mas sem nenhum tipo de modificação nos valores, causando situações como na Figura 3.3 e na Figura 3.4.

Figura 3.3: Análise da tabela de banco de dados do BlueWave

ID	UTCTimestamp_Ticks	LogType	NotSync	Node1_USERGVL1_var1	_Node1_USERGVL1_var1_Q	Node1_USERGVL1_var2	_Node1_USERGVL1_var2_Q	Node1_USERGVL1_var3
0	636643328297644726	0	0	0	192	0	192	0
1	636643328592057552	1	0	3136615	192	3136615	192	3136615
2	636643328597674135	1	0	440741	192	440741	192	440741
3	636643328598091430	1	0	440741	192	440741	192	440741
4	636643328598600088	1	0	440741	192	440741	192	440741
5	63664332859917389	1	0	440741	192	440741	192	440741
6	636643328600466535	1	0	440741	192	440741	192	440741
7	636643328601204194	1	0	440741	192	440741	192	440741
8	636643328601841765	1	0	440741	192	440741	192	440741

Na Figura 3.3, é possível notar que da linha 3 até o fim da Figura 3.3 os valores das variáveis se repetem.

Figura 3.4: Análise da tabela de banco de dados do BlueWave

ID	UTCTimestamp_Ticks	LogType	NotSync	Node1_USERGVL2_var1001	_Node1_USERGVL2_var1001_Q	Node1_USERGVL2_var1002	_Node1_USERGVL2_var1002_Q	Node1_USERGVL2_var1003
0	636643328269619700	0	0	0	192	0	192	0
1	636643328670587921	1	0	1867387	192	1867387	192	1867387
2	636643329092328242	1	0	1930164	192	1930164	192	1930164
3	636643329173885953	1	0	5012465	192	5012465	192	5012465
4	636643329627891687	2	0	5012465	192	5012465	192	5012465
5	636643340060402539	0	0	0	192	0	192	0
6	636643340943874303	1	0	185999	192	185999	192	185999
7	636643346287380426	2	0	185999	192	185999	192	185999
8	636643347358471986	0	0	0	192	0	192	0
9	636643348098021904	1	0	528334	192	528334	192	528334

Na Figura 3.4, é possível notar que algumas linhas estão repetidas, como 4 e 5 e também 7 e 8.

Enquanto as outras tabelas, que apresentam funcionamento esperado, estão criando um registro para cada coleta, a tabela com funcionamento fora dos padrões está criando múltiplos registros, o que acaba gerando um uso desnecessário do espaço em disco. Esse defeito encontrado pode ser a causa do uso excessivo de disco verificado pelos clientes do *BlueWave*.

4 CONCLUSÃO

Testes de desempenho são capazes de mostrar os gargalos do sistema e auxiliar a detecção de possíveis problemas que outros tipos de testes não são capazes de encontrar. A formulação do plano de teste tem que ter claro qual o objetivo do mesmo, afinal, é preciso minimizar o número de variáveis para que o gargalo possa ser apontado.

Neste trabalho foi feita a descrição de um ambiente de automação industrial, além da criação e execução de um plano de teste de desempenho em um dos *softwares* que faz parte desse sistema, o SCADA e, em especial, o *BlueWave*. Foi feita a escolha de utilizar a comparação com outro *software* da indústria devido às reclamações dos clientes. Também não foi possível localizar versões anteriores do *BlueWave* que apresentassem mudanças no historiador para que pudesse ser feita uma comparação. A opção de modificar todas as variáveis da tabela todas as vezes pode ter sido decisória para que o *Ignition* desempenhasse melhor, pois era seu pior caso.

A partir dos testes realizados nos historiadores do sistemas SCADA *Ignition* e *BlueWave*, foi possível verificar que os dois sistemas apresentam algumas deficiências na forma de uso de espaço no disco. Para esse trabalho, foi analisado apenas o *BlueWave*.

No teste do pior caso do sistema, no qual apenas um ponto de coleta tem seu valor modificado, é possível concluir que o *BlueWave* faz um uso de espaço em disco menor que seu concorrente, Isso ocorre devido à forma que as tabelas foram construídas, utilizando colunas para representar as variáveis e suas informações.

No teste do melhor caso com apenas uma inserção, o *BlueWave* e o *Ignition* ocuparam o mesmo espaço.

Nos testes com diversas inserções, foi possível verificar que o sistema ocupou um espaço consideravelmente menor em disco quando comparado ao *Ignition*. Como em todas as operações todas as *tags* tinham seu valor modificado, isso pode ter atrapalhado o sistema de compressão de dados do *Ignition* e, como o cenário representava o melhor caso do *BlueWave*, que é um dos cenários típicos de uso real, o espaço utilizado pelos dados em suas tabelas representou aproximadamente um terço do utilizado pelo concorrente.

Durante os testes iniciais com o banco de dados, foi possível perceber que o *BlueWave* tinha um comportamento peculiar em algumas tabelas, que tinham um número muito maior de registros apesar de terem sido performados o mesmo número de inserções em todas. Essa característica pode ser vista em diferentes cenários criados, levando à conclusão que é um problema que se repete em diferentes projetos, apesar das *tags* serem

criadas da mesma forma.

Pode-se concluir que o uso excessivo de disco quando comparado aos seus concorrentes atestado pelos clientes que utilizam o *BlueWave* pode ser causado pela criação de tabelas nas quais poucos pontos de coleta variam juntos, o que leva à grande repetição de informações nos registros ou os projetos criados apresentam uma ou mais tabelas com o defeito de inserção de dados.

Para determinar a causa exata e tomar a decisão de como corrigi-la, seria preciso fazer um levantamento dos cenários de uso reais do sistema, informação à qual não foi possível ter acesso, e compreender como são compostas as tabelas pelos engenheiros para que seja possível averiguar. Como cada empresa que utiliza o sistema tem a liberdade de criar suas tabelas, teria que ser determinado um subconjunto para ser avaliado e encontrar a melhor solução.

4.1 Trabalhos Futuros

A partir das conclusões desse trabalho, ficam abertos os caminhos para futuros trabalhos.

O primeiro seria uma análise das tabelas criadas por engenheiros reais, bem como a frequência na qual as *tags* tem seu valor alterado para que seja possível determinar como uma potencial refatoração da forma que as tabelas são criadas no banco de dados pode ocorrer e como seria mais proveitoso para os engenheiros e técnicos que utilizam o sistema, já que assim o espaço usado em disco seria reduzido, dando maior longevidade ao sistema sem a necessidade de trocas de *hardware*

O segundo seria uma refatoração do *BlueWave* para corrigir o problema que faz com que muitos registros sejam inseridos na tabela. Essa falha se mostrou extremamente nociva ao desempenho do sistema em relação ao uso de espaço em disco, criando inserção de informações desnecessárias. Como falado na seção 2.3, é preciso ver se o custo de realizar o *log* dos eventos compensa pelo valor da informação e, nesse caso, o custo é extremamente alto.

Uma terceira possibilidade seria o *benchmark* de outros sistemas SCADA na indústria que têm alta aprovação de seus usuários para que seja possível determinar onde o *BlueWave* se encaixa no uso de disco em comparação com mais *softwares* para que se tenha uma ideia real se existe a necessidade de melhora de desempenho do mesmo.

REFERÊNCIAS

- APACHE. **Log Files**. Apache, 2018. Acessado em 2018-06-22. Disponível em: <<https://httpd.apache.org/docs/1.3/logs.html>>.
- AUTOMATION, I. **Historian - Ignition User Manual 7.9**. 2018. Acessado em 2018-07-13. Disponível em: <<https://docs.inductiveautomation.com/display/DOC79/>>.
- AUTOMATION, I. **Introducing Ignition - Ignition User Manual 7.9**. 2018. Acessado em 2018-07-16. Disponível em: <<https://docs.inductiveautomation.com/display/DOC79/IntroducingIgnition>>.
- AUTOMATION, I. **System Architectures - Ignition User Manual 7.9**. Inductive Automation, 2018. Acessado em 2018-05-16. Disponível em: <<https://docs.inductiveautomation.com/display/DOC79/SystemArchitectures>>.
- AUTOMAÇÃO, A. S. de. **Manual de Usuário do BlueWave**. 2012.
- AUTOMAÇÃO, A. S. de. **Manual de Utilização**. 2012. Acessado em 2018-10-26. Disponível em: <<https://forum.inductiveautomation.com/t/disk-space-calculation/4972>>.
- BEAL, V. **SQL - structured query language**. Acessado em 2018-10-02. Disponível em: <<https://www.webopedia.com/TERM/S/SQL.html>>.
- CIGNO, R. L. **Systems – Common Mistakes – PE tools**. [S.l.]: University of Trento, 2017.
- COLCHER, S. **Modelagem Analítica do Desempenho de Sistemas de Computação Modelagem Analítica do Desempenho de Sistemas**. PUC-RIO, 2009. Acessado em 2018-08-10. Disponível em: <<http://www.inf.puc-rio.br/~inf2511/>>.
- CONTROLS, I. J. **Commissioning AspenTech® InfoPlus.21™ Historian**. [S.l.]: Johnson Controls, Inc., 2001. Acessado em 2018-03-28.
- CORPORATION, I. **Creating tables for staging data**. IBM Corporation, 2018. Acessado em 2018-05-18. Disponível em: <https://www.ibm.com/support/knowledgecenter/SSEPGG_9.5.0/com.ibm.db2.luw.admin.dboj.doc/doc/t0006862.html>.
- CORPORATION, I. **Logs**. IBM Corporation, 2018. Acessado em 2018-04-23. Disponível em: <https://www.ibm.com/support/knowledgecenter/en/SSMKHH_9.0.0/com.ibm.etools.mft.doc/au14160_.htm>.
- CORPORATION, I. **Message logs**. IBM Corporation, 2018. Acessado em 2018-09-17. Disponível em: <https://www.ibm.com/support/knowledgecenter/en/SSNGTE_7.1.0/com.ibm.tspm.doc_7.1/probdet/messagelogs.html>.
- CORPORATION, I. **System Workload**. IBM Corporation, 2018. Acessado em 2018-07-06. Disponível em: <https://www.ibm.com/support/knowledgecenter/en/ssw_aix_72/com.ibm.aix.performance/sys_workload.htm>.

CORPORATION, M. **Event Logging**. Microsoft Corporation, 2014. Acessado em 2018-04-02. Disponível em: <[https://msdn.microsoft.com/en-us/library/windows/desktop/aa363652\(v=vs.85\).aspx](https://msdn.microsoft.com/en-us/library/windows/desktop/aa363652(v=vs.85).aspx)>.

CORPORATION, M. **Message Logging**. Microsoft Corporation, 2017. Acessado em 2018-05-28. Disponível em: <<https://docs.microsoft.com/en-us/dotnet/framework/wcf/diagnostics/message-logging>>.

CORPORATION, N. I. **Serial Communication Overview**. National Instruments Corporation, 2012. Acessado em 2018-07-20. Disponível em: <<http://www.ni.com/white-paper/2895/en/>>.

EXCELENCE, A. **What is SCADA?** Automation Excellence Private Limited, 2018. Disponível em: <<http://www.automationexcellence.com/scada.html>>.

FOUNDATION, O. **Classic Data Access- OPCDA**. OPC Foundation, 2018. Acessado em 2018-08-10. Disponível em: <<https://opcfoundation.org/developer-tools/specifications-classic/data-access/>>.

FU, Q. et al. **Where do Developers Log? An Empirical Study on Logging Practices in Industry**. [S.l.: s.n.], 2014. 24–33 p.

HOLLIFIELD, B. **Understanding and Applying the ANSI/ ISA 18.2 Alarm Management Standard**. [S.l.]: PAS GLobal, 2010.

HUNTER, B. **benchmark**. Thistledown Press, 1982. Acessado em 2018-03-09. Disponível em: <<https://en.oxforddictionaries.com/definition/benchmark>>.

INCAPSULA, I. **DISTRIBUTED DENIAL OF SERVICE ATTACKS - WHAT DOES DDOS MEAN?** Imperva Inc., 2018. Acessado em 2018-09-08. Disponível em: <<https://www.incapsula.com/ddos/denial-of-service.html>>.

INSTRUMENTS, N. **RS-232, RS-422, RS-485 Serial Communication General Concepts**. [S.l.], 2018.

JAIN, R. **The art of computer systems performance analysis - techniques for experimental design, measurement, simulation, and modeling**. [S.l.]: Wiley, 1991. I-XXVII, 1-685 p. (Wiley professional computing). ISBN 978-0-471-50336-1.

KUROSE, J. F.; ROSS, K. W. **Computer Networking: A Top-Down Approach (6th Edition)**. 6th. ed. [S.l.]: Pearson, 2012. ISBN 0132856204, 9780132856201.

LLC, D. **Dynatrace Performance Benchmarks: Methodology**. Waltham, Massachusetts, 2017. 9 p. Acessado em 2018-08-19.

LLC, I. A. **Historian - Ignition User Manual 7.9**. 2018. Acessado em 2018-07-13. Disponível em: <<https://docs.inductiveautomation.com/display/DOC79/Historian>>.

M.M., A. I. O. T. Architecture of industrial automation systems. **European Scientific Journal**, European Scientific Journal, v. 10, Jan 2014. ISSN 1857- 7431. Disponível em: <<https://eujournal.org/index.php/esj/article/viewFile/2628/2489>>.

OGATA, K. **Modern Control Engineering**. 4th. ed. Upper Saddle River, NJ, USA: Prentice Hall PTR, 2001. ISBN 0130609072.

ORGANIZATION, I. M. **Modbus Specification and Implementation Guide**. 2018. Acessado em 2018-05-26. Disponível em: <<http://www.modbus.org/specs.php>>.

ORNELAS, L. V. **Importance of Event Logging**. [S.l.], 2003.

OSISOFT. **PI System Architecture, Planning and Implementation**. [S.l.]: OsiSoft, 2016.

PRODUCTS, A. S. A. **MicroSCADA Pro SYS 600 9.2 - System Configuration**. [S.l.]: ABB Group, 2014. Acessado em 2018-03-30.

PUROMÄKI, T. **Data Validation in SCADA System**. 2010. Tese da VAASAN AMMATTIKORKEAKOULU, Vaasa, Finlândia.

SKORKOVSKÝ. **BASIC ERP ARCHITECTURE**. [S.l.]: Masaryk University, 2011.

SPLUNK, I. **SIEM, AIOps, Application Management, Log Management, Machine Learning, and Compliance**. 2001. Acessado em 2018-04-20. Disponível em: <<https://www.splunk.com/>>.

SYSTEMS, I. I. **Wonderware FactorySuite: IndustrialSQL Server Historian Concepts Guide**. Lake Forest, CA, 2005. 216 p. Acessado em 2018-04-02. Disponível em: <<http://platforma.astor.com.pl/files/getfile/id/3781>>.

TECHNOLOGY, E. **What is Industrial Automation | Types of Industrial Automation**. Electrical Technology, 2015. Acessado em 2018-08-28. Disponível em: <<https://www.electricaltechnology.org/2015/09/what-is-industrial-automation.html>>.

TERGOLINA, R. L. **Sensores Industriais**. [S.l.]: PUCRS, 2015.

TIPS, M. **What is the transaction log?** 2014. Acessado em 2018-05-30. Disponível em: <<https://www.mssqltips.com/sqlservertutorial/3302/what-is-the-transaction-log/>>.

ZENG, M. L.; QIN, J. **Metadata**. [S.l.]: Facet Publishing, 2016.

ZHIHONG, L.; PEARSON, S. An inside look at industrial ethernet communication protocols. Texas Instruments, Dallas, Texas, USA, 2017.

APÊNDICE A — RELATÓRIO GERADO PELO *MICROSOFT SQL SERVER* 2014 EXPLICITANDO O USO DE DISCO POR CADA TABELA NO ESTADO INICIAL E APÓS A INSERÇÃO DE UM ITEM

Como indicado na definição de plano de teste, é necessário ter algo para comparar com o resultado dos testes e poder determinar quanto espaço foi utilizado pelas operações. Para isso, abaixo, está o estado inicial do banco de dados.

A tabela utilizada para os testes do BW é a *dbo.DBTesting1*.

Figura A.1: Uso de disco das tabelas do Ignition antes do início dos testes

Nome da Tabela	Nº de Registros	Reservado (KB)	Dados (KB)	Índices (KB)	Não Usado (KB)
<i>dbo.sql_data_1_2018_06</i>	44.804	4.432	3.176	1.224	32
<i>dbo.sqlh_drv</i>	2	16	8	8	0
<i>dbo.sqlh_partitions</i>	1	16	8	8	0
<i>dbo.sqlh_sce</i>	1	48	8	40	0
<i>dbo.sqlh_scinfo</i>	1	16	8	8	0
<i>dbo.sqlh_te</i>	20.152	2.512	1.376	1.104	32

Figura A.2: Uso de disco das tabelas do BlueWave antes do início dos testes

Nome da Tabela	Nº de Registros	Reservado (KB)	Dados (KB)	Índices (KB)	Não Usado (KB)
<i>dbo.DBTesting1</i>	12	168	96	48	24
<i>dbo.DBTesting2</i>	12	168	96	48	24
<i>dbo.DBTesting3</i>	449	3.720	3.592	80	48
<i>dbo.DBTesting4</i>	12	168	96	48	24
<i>dbo.DBTesting5</i>	187	1.576	1.496	48	32
<i>dbo.Table1</i>	0	0	0	0	0

Atualização de um valor da tabela

Figura A.3: Uso de disco das tabelas do Ignition após uma tag ter seu valor modificado

Nome da Tabela	Nº de Registros	Reservado (KB)	Dados (KB)	Índices (KB)	Não Usado (KB)
<i>dbo.sql_data_1_2018_06</i>	44.805	4.432	3.176	1.224	32
<i>dbo.sqlh_drv</i>	2	16	8	8	0
<i>dbo.sqlh_partitions</i>	1	16	8	8	0
<i>dbo.sqlh_sce</i>	1	48	8	40	0
<i>dbo.sqlh_scinfo</i>	1	16	8	8	0
<i>dbo.sqlh_te</i>	20.152	2.512	1.376	1.104	32

Figura A.4: Uso de disco das tabelas do BlueWave após uma tag ter seu valor modificado

Nome da Tabela	Nº de Registros	Reservado (KB)	Dados (KB)	Índices (KB)	Não Usado (KB)
<i>dbo.DBTesting1</i>	14	168	112	48	8
<i>dbo.DBTesting2</i>	12	168	96	48	24
<i>dbo.DBTesting3</i>	449	3.720	3.592	80	48
<i>dbo.DBTesting4</i>	12	168	96	48	24
<i>dbo.DBTesting5</i>	187	1.576	1.496	48	32
<i>dbo.Table1</i>	0	0	0	0	0

**APÊNDICE B — RELATÓRIO GERADO PELO *MICROSOFT SQL SERVER*
2014 EXPLICITANDO O USO DE DISCO POR CADA TABELA APÓS A
MODIFICAÇÃO DE TODOS OS PONTOS DE COLETA UMA VEZ**

O estado inicial dos testes abaixo é o estado descrito no Apêndice A.

A tabela utilizada para os testes do BW é a *dbo.DBTesting1*.

Figura B.1: Uso de disco das tabelas do Ignition após todas as tags terem seu valor modificado

Nome da Tabela	Nº de Registros	Reservado (KB)	Dados (KB)	Índices (KB)	Não Usado (KB)
<i>dbo.sqlt_data_1_2018_06</i>	45.306	4.496	3.208	1.240	48
<i>dbo.sqlth_drv</i>	2	16	8	8	0
<i>dbo.sqlth_partitions</i>	1	16	8	8	0
<i>dbo.sqlth_sce</i>	1	48	8	40	0
<i>dbo.sqlth_scinfo</i>	1	16	8	8	0
<i>dbo.sqlth_te</i>	20.152	2.512	1.376	1.104	32

Figura B.2: Uso de disco das tabelas do BlueWave após todas as tags terem seu valor modificado

Nome da Tabela	Nº de Registros	Reservado (KB)	Dados (KB)	Índices (KB)	Não Usado (KB)
<i>dbo.DBTesting1</i>	18	232	144	48	40
<i>dbo.DBTesting2</i>	12	168	96	48	24
<i>dbo.DBTesting3</i>	449	3.720	3.592	80	48
<i>dbo.DBTesting4</i>	12	168	96	48	24
<i>dbo.DBTesting5</i>	187	1.576	1.496	48	32
<i>dbo.Table1</i>	0	0	0	0	0

APÊNDICE C — RELATÓRIO GERADO PELO *MICROSOFT SQL SERVER* 2014 EXPLICITANDO O USO DE DISCO POR CADA TABELA APÓS OS TESTES DE LONGA DURAÇÃO

- *Cenário Inicial:* Antes dos testes começarem, foi gerado um relatório de uso de memória pelas tabelas¹ criadas pelos historiadores dos sistemas para que fosse possível comparar os espaços utilizados. Os resultados estão na Figura C.1 e Figura C.2.

A tabela utilizada para os testes do BW é a `dbo.DBTesting1`.

Figura C.1: Uso de disco das tabelas do BlueWave antes de todas as tags terem seu valor modificado diversas vezes

Nome da Tabela	Nº de Registros	Reservado (KB)	Dados (KB)	Índices (KB)	Não Usado (KB)
<code>dbo.DBTesting1</code>	732	5.968	5.856	112	0
<code>dbo.DBTesting2</code>	12	168	96	48	24
<code>dbo.DBTesting3</code>	449	3.720	3.592	80	48
<code>dbo.DBTesting4</code>	12	168	96	48	24
<code>dbo.DBTesting5</code>	187	1.576	1.496	48	32
<code>dbo.Table1</code>	0	0	0	0	0

Figura C.2: Uso de disco das tabelas do Ignition antes de todas as tags terem seu valor modificado diversas vezes

Nome da Tabela	Nº de Registros	Reservado (KB)	Dados (KB)	Índices (KB)	Não Usado (KB)
<code>dbo.sqlt_data_1_2018_06</code>	317.221	30.928	20.800	10.072	56
<code>dbo.sqlth_drv</code>	2	16	8	8	0
<code>dbo.sqlth_partitions</code>	1	16	8	8	0
<code>dbo.sqlth_sce</code>	3	48	8	40	0
<code>dbo.sqlth_scinfo</code>	1	16	8	8	0
<code>dbo.sqlth_te</code>	20.152	2.512	1.376	1.104	32

- *Primeira Rodada:* Após 30 minutos de atualizações de valores a cada 10 segundos, as tabelas estavam utilizando o espaço abaixo:

¹A geração desse relatório é uma ferramenta do próprio *Microsoft SQL Server 2014*.

Figura C.3: Uso de disco das tabelas do BlueWave após todas as tags terem seu valor modificado diversas vezes - Rodada 1

Nome da Tabela	Nº de Registros	Reservado (KB)	Dados (KB)	Índices (KB)	Não Usado (KB)
dbo.DBTesting1	921	7.512	7.368	128	16
dbo.DBTesting2	12	168	96	48	24
dbo.DBTesting3	449	3.720	3.592	80	48
dbo.DBTesting4	12	168	96	48	24
dbo.DBTesting5	187	1.576	1.496	48	32
dbo.Table1	0	0	0	0	0

Figura C.4: Uso de disco das tabelas do Ignition após todas as tags terem seu valor modificado diversas vezes - Rodada 1

Nome da Tabela	Nº de Registros	Reservado (KB)	Dados (KB)	Índices (KB)	Não Usado (KB)
dbo.sqlt_data_1_2018_06	406.478	37.904	24.880	12.976	48
dbo.sqlth_drv	2	16	8	8	0
dbo.sqlth_partitions	1	16	8	8	0
dbo.sqlth_sce	4	48	8	40	0
dbo.sqlth_scinfo	1	16	8	8	0
dbo.sqlth_te	20.152	2.512	1.376	1.104	32

- *Segunda Rodada:* Utilizando as tabelas do relatório anterior como ponto de partida, novamente um teste de 30 minutos de atualizações de valores a cada 10 segundos foi rodado e, ao final, as tabelas estavam utilizando o espaço abaixo:

Figura C.5: Uso de disco das tabelas do BlueWave após todas as tags terem seu valor modificado diversas vezes - Rodada 2

Nome da Tabela	Nº de Registros	Reservado (KB)	Dados (KB)	Índices (KB)	Não Usado (KB)
dbo.DBTesting1	1.104	8.992	8.832	136	24
dbo.DBTesting2	12	168	96	48	24
dbo.DBTesting3	449	3.720	3.592	80	48
dbo.DBTesting4	12	168	96	48	24
dbo.DBTesting5	187	1.576	1.496	48	32
dbo.Table1	0	0	0	0	0

Figura C.6: Uso de disco das tabelas do Ignition após todas as tags terem seu valor modificado diversas vezes - Rodada 2

Nome da Tabela	Nº de Registros	Reservado (KB)	Dados (KB)	Índices (KB)	Não Usado (KB)
dbo.sqlt_data_1_2018_06	495.978	45.008	29.048	15.904	56
dbo.sqlth_drv	2	16	8	8	0
dbo.sqlth_partitions	1	16	8	8	0
dbo.sqlth_sce	4	48	8	40	0
dbo.sqlth_scinfo	1	16	8	8	0
dbo.sqlth_te	20.152	2.512	1.376	1.104	32

- *Terceira Rodada:* Assim como nos itens anteriores, o mesmo processo foi repetido e os espaço ocupado pelas tabelas do Ignition e do BlueWave, respectivamente, foram:

Figura C.7: Uso de disco das tabelas do BlueWave após todas as tags terem seu valor modificado diversas vezes - Rodada 3

Nome da Tabela	Nº de Registros	Reservado (KB)	Dados (KB)	Índices (KB)	Não Usado (KB)
dbo.DBTesting1	1.287	10.472	10.296	152	24
dbo.DBTesting2	12	168	96	48	24
dbo.DBTesting3	449	3.720	3.592	80	48
dbo.DBTesting4	12	168	96	48	24
dbo.DBTesting5	187	1.576	1.496	48	32
dbo.Table1	0	0	0	0	0

Figura C.8: Uso de disco das tabelas do Ignition após todas as tags terem seu valor modificado diversas vezes - Rodada 3

Nome da Tabela	Nº de Registros	Reservado (KB)	Dados (KB)	Índices (KB)	Não Usado (KB)
dbo.sqlt_data_1_2018_06	585.544	52.112	33.264	18.808	40
dbo.sqlth_drv	2	16	8	8	0
dbo.sqlth_partitions	1	16	8	8	0
dbo.sqlth_sce	4	48	8	40	0
dbo.sqlth_scinfo	1	16	8	8	0
dbo.sqlth_te	20.152	2.512	1.376	1.104	32

**APÊNDICE D — TABELA COM VALORES EM KB DESCRIVENDO ESPAÇO
DE USO EM DISCO APÓS OS TESTES DE LONGA DURAÇÃO**

Tabela D.1: Tabela explicitando uso de disco dos historiadores *BlueWave* e *Ignition* em cada uma das rodadas dos testes de longa duração(em Kb)

	BlueWave	Ignition
Início	0	0
Rodada 1	1512	4080
Rodada 2	2976	8248
Rodada 3	4440	12464