



ESTUDO COMPARATIVO ENTRE FERRAMENTAS DE SUPERVISÃO, CONTROLE E AQUISIÇÃO DE DADOS E A IMPORTÂNCIA DESTAS PARA O ENSINO EM ENGENHARIA

Mateus Hufnagel Maranha de Faria – mateushufnagel@gmail.com

Marlon Ramos Silva – marlonramoss@yahoo.com.br

Ângelo Rocha de Oliveira – angelo@leopoldina.cefetmg.br

Marlon José do Carmo – marlon@leopoldina.cefetmg.br

Lindolpho Oliveira de Araújo Júnior – lindolpho@leopoldina.cefetmg.br

Centro Federal de Educação Tecnológica de Minas Gerais, *Campus III*

Rua José Peres, 558 – Centro

36700-000 – Leopoldina – MG

Resumo: *Este artigo apresenta um estudo comparativo entre dois softwares de supervisão, controle e aquisição de dados, e a importância dessas ferramentas no ensino em engenharia. Para realizar este estudo, escolheram-se dois softwares bastante utilizados nacionalmente em diversas áreas da automação, que são o Elipse SCADA e o ScadaBR, aos quais foram aplicados parâmetros como ergonomia de software, protocolos de comunicação com equipamentos de campo, sistemas de banco de dados, segurança dos dados e aplicações, dentre outros. Destaca-se também que trabalhando com estas ferramentas, a construção do conhecimento e a prática interdisciplinar consolidam o aprendizado dos fundamentos teóricos, trazendo-os através de aplicações simuladas de situações reais.*

Palavras-chave: *Engenharia de controle e automação, Mecatrônica, SCADA, Construtivismo, Ergonomia de Software.*

1. INTRODUÇÃO

O mundo evolui de forma vertiginosa. Diariamente deparamos com a alteração parcial ou total das condições de uma situação no nível micro ou macro, com mudanças rápidas e crescentes, que modificam de alguma maneira a realidade (IÓRIO, 2002).

A economia mundial foi marcada no século XVIII pela invenção da máquina a vapor; no fim do século XX, os propulsores da nova revolução do desenvolvimento foram – e continuam a sê-lo – a tecnologia (representada pela informática e pelo aperfeiçoamento dos transportes e das comunicações) e a globalização (IÓRIO, 2002).

Em uma sociedade moderna, sistemas de controle são itens essenciais para um bom funcionamento de indústrias e laboratórios de diversas áreas. Um sistema de controle consiste em subsistemas e processos (ou plantas) construídos com o objetivo de se obter uma saída desejada com desempenho desejado, para uma entrada específica fornecida (NISE, 2011).

Os sistemas que se caracterizam pelo gerenciamento de processos de forma integrada, em

Realização:



Organização:





geral, são designados pelo nome de Manufatura Integrada por Computador, ou simplesmente CIM (*Computer Integrated Manufacturing*) (ALBUQUERQUE & ALEXANDRIA, 2009). Atualmente, a base de um CIM é formada por SDCD – Sistema Digital de Controle distribuído – que representa todos os níveis de controle e execução dos processos. É relevante destacar a importância dos programas de aquisição de dados, supervisão e controle (SCADA), das redes de comunicação que utilizam protocolos industriais (*fieldbus*), pois são essenciais para o funcionamento dos sistemas CIM e SDCD (ALBUQUERQUE & ALEXANDRIA, 2009).

O cenário industrial atual é cercado por equipamentos e máquinas com o objetivo principal de medir e controlar os diversos tipos de processos físicos, químicos, entre outros. Dentro do grupo destes equipamentos, encontram-se sensores, atuadores e controladores lógicos programáveis que oferecem alguma Interface Digital para conexão com computador. Um sensor pode ser definido como um transdutor que altera a sua característica física interna devido a um fenômeno físico externo (ROSÁRIO, 2005). Do ponto de vista tecnológico, os dispositivos mecânicos que aplicam ou fazem atuar energia mecânica sobre uma máquina, para realizar determinado trabalho, são chamados de atuadores (DISTEFANO, 1978). No ensino da mecatrônica, à guisa de exemplo, podemos utilizar componentes LEGO™ na montagem de dispositivos mecatrônicos, o que permite fácil visualização na formação educacional e profissional dessa área (ROSÁRIO, 2005).

Ao longo deste artigo, será dado foco na utilização e comparação entre dois sistemas SCADA, bem conhecidos nacionalmente, visando sua utilização no ensino em engenharia de controle e automação, uma vez que, pode-se observar através das informações contidas na “Figura 1”, sistemas SCADA são largamente utilizados em diferentes setores da pirâmide de automação e controle de um sistema, tornando sua utilização na graduação de grande importância para que haja uma melhor associação entre as disciplinas de áreas diferentes (mecânica, eletrônica, informática).

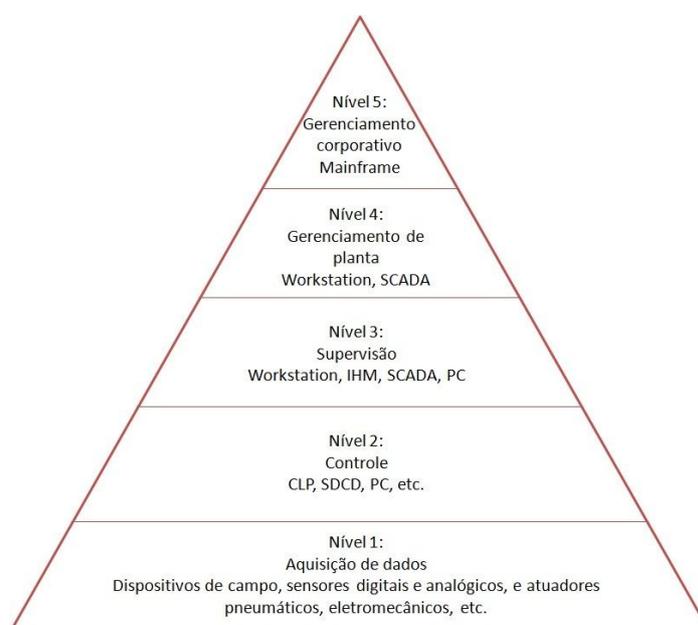


Figura 1 - Pirâmide de automação com sistema SCADA.



As realizações de trabalhos multidisciplinares, além de aumentar o interesse do estudante em cada disciplina, separadamente, cooperam para que o conteúdo das mesmas seja fixado de maneira construtivista. Nessa concepção epistemológica o professor tem a tarefa principal de monitorar o crescimento cognitivo e o amadurecimento pessoal dos estudantes, contribuindo para a construção, por parte de cada um, de um conhecimento científico pessoal, com a dupla característica de ser semelhante ao conhecimento científico estabelecido e ter continuidade com a própria ecologia conceitual (STRIKE & POSNER, 1992).

O presente trabalho tem como finalidade a análise comparativa entre um software proprietário, Eclipse SCADA e outro de licença livre, expondo suas características mais importantes com o intuito de utilizá-los em laboratórios na graduação em engenharia.

O artigo está dividido da seguinte forma: apresenta-se na seção 2 uma contextualização da utilização dos softwares do tipo SCADA e alguns exemplos de softwares livres conhecidos; na seção 3 é dada uma introdução sobre a ferramenta ScadaBR; na seção 4 é apresentado o Eclipse SCADA; na seção 5, é feito um processo comparativo entre o ScadaBR e Eclipse SCADA, onde são utilizados métodos como ergonomia, protocolos de comunicação, linguagem de programação, bancos de dados, segurança e extensibilidade, assim como a importância de cada um desses parâmetros; na seção 6 podem ser verificadas as considerações finais sobre o estudo realizado neste artigo.

2. SOFTWARES DO TIPO SCADA

No mercado atual, existem muitos softwares SCADA disponíveis em diversas línguas, com custo relativamente elevado, o que gera uma inviabilização de sua utilização para automação de sistemas de menor porte ou em instituições de ensino tecnológico. Contudo, do decorrer dos anos 90, surgiram softwares livres com algumas funcionalidades de SCADA, que acarretou no incentivo de desenvolvimento de novas tecnologias baseadas em código aberto.

2.1. Visual

O Visual é um software antigo, amplamente utilizado no estudo de SCADA. Possui um conjunto de programas com o objetivo de controlar, operar e monitorar máquinas industriais através de um computador, usando um computador local ou através da internet.

2.2. Beremiz

Beremiz é um software livre de código aberto, em conformidade com IEC-61131 entre outras normas. É definido como um ambiente de desenvolvimento integrado para automação de máquinas.

O projeto Beremiz, foi desenvolvido a partir de contribuições principalmente da França e de Portugal, e seu funcionamento vai além das funcionalidades de um simples SCADA. Uma das ferramentas que chama bastante atenção no Beremiz é sua interface baseada em SVG (que possibilita realizar animações a partir de desenhos vetoriais).

2.3. openSCADA

O openSCADA é um software livre que roda em várias plataformas operacionais. É flexível e seguro, o que garante que um conjunto de ferramentas possam ser combinadas de



diversas maneiras diferentes. Porém, o openSCADA exige que seu operador detenha algum conhecimento de programação adquirido anteriormente.

2.4. Mango

O Mango é um software livre de código aberto canadense que possui muitas ferramentas importantes em sistemas SCADA, desde diversos barramentos ou protocolos de comunicação, passando por sensores variados, até a construção de telas interativas. A tecnologia web caracteriza a base de funcionamento do Mango, sendo que suas interfaces são visualizadas a partir de browsers.

2.5. Likindoy

Likindoy detém uma arquitetura voltada para operação em rede. Ele foi desenvolvido principalmente para sistemas geograficamente distribuídos como, por exemplo, distribuição de gás e saneamento básico.

3. SCADABR

O ScadaBR é um software gratuito de código aberto, criado através de uma parceria entre várias instituições (CNPq, Finep, Sebrae, Conetec, MCA, Fundação Certi e Unis Sistemas), onde hoje, seu desenvolvimento é realizado também por qualquer usuário que pertença à comunidade do ScadaBR, que é sem fins lucrativos. Voltado para a Aquisição de Dados e Controle Supervisório, é bastante utilizado no desenvolvimento de aplicações automatizadas para diferentes ambientes de trabalho: laboratório, automação predial, saneamento, sistemas de energia, etc.

A “Figura 2” mostra uma enquete levantada pelo portal oficial do ScadaBR entre 2011 a 2012, feita com 121 usuários do software.

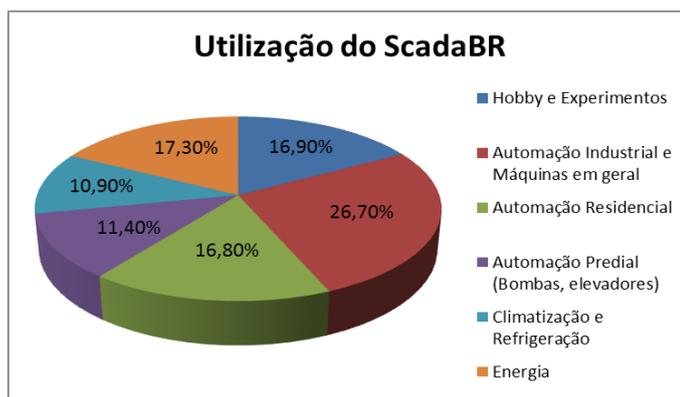


Figura 2 – Gráfico “Aplicações ScadaBR”.

Dentre as principais funcionalidades do ScadaBR tem-se: Comunicação em mais de 20 protocolos incluindo: OPC, Modbus, ASCII, Bacnet, dentre outros. Gera gráficos e relatórios, alarmes e lógicas programáveis e é disponível em versões Linux e Windows (XP e 7).



4. ELIPSE SCADA

O Elipse SCADA é um software desenvolvido pela Elipse Software, utilizado na criação de aplicativos de supervisão e controle nas mais diversas áreas. É baseado na tríade da acessibilidade, amabilidade e flexibilidade. Desta forma, ele é uma ferramenta ideal para a automação, pois consegue eliminar a necessidade de soluções demoradas e relativamente caras, garantindo certa competitividade, eficiência e qualidade para o processo. Também está disponível para basicamente todas as versões da plataforma Microsoft Windows (98, ME, NT, 2000, XP, 2003, Vista, 7) e Windows CE.

5. SCADABR X ELIPSE SCADA

Tem-se como principal objetivo ao comparar às duas ferramentas do tipo SCADA, a otimização do estudo multidisciplinar em laboratório. Desta forma, realizamos um levantamento das principais funções destas ferramentas, que são importantes para o ensino na graduação.

5.1. Ergonomia

Praticamente todos os países, hoje em dia, dependem de complexos sistemas com base em computadores. Cada vez mais os produtos incorporam de algum modo, computadores e software de controle (SOMMERVILLE, 2003).

Nesses sistemas, o software representa uma grande e crescente proporção do custo total do sistema. Por isso, há crescente necessidade de se produzir software, de modo que apresentem boa relação custo-benefício (SOMMERVILLE, 2003).

A Ergonomia de Software apresenta grande importância dentro da Engenharia de Software, que é uma disciplina da engenharia que se ocupa de todos os aspectos da produção de software, desde os estágios iniciais de especificação do sistema até a manutenção desse sistema, depois que ele entrou em operação (SOMMERVILLE, 2003).

A “Tabela 1” apresenta o comportamento da Ergonomia de Software no ScadaBR e no Elipse SCADA. A tabela foi construída a partir dos princípios básicos que permeiam a Ergonomia de Software, que foram definidas pelo professor Roberto Cabral de Mello Borges (BORGES, 2012), M.Sc. em Ciência da Computação pela Universidade Federal do Rio Grande do Sul.

Tabela 1 - Comparação dos parâmetros de ergonomia entre ScadaBR e Elipse SCADA

Princípios Básicos	ScadaBR	Elipse SCADA
1 – Esforço mínimo do usuário.	X	X
2 – Memória mínima do usuário.	-	X
3 – Frustração mínima.	X	X
4 – Maximizar o uso de padrões e hábitos.	X	X
5 – Máxima tolerância para diferenças humanas.	X	X
6 – Notificação imediata de problemas	X	X
7 – Controle máximo de tarefas pelo usuário.	X	X
8 – Apoio máximo às tarefas.	-	-



1 – Esforço mínimo do usuário: refere-se ao fato de um programa poder ser reutilizável, assim como rotinas, consultas, comandos, etc. Além disso, o software deve garantir uma facilidade ao acesso de informações sobre o sistema (manual do software ou ícone “help”).

2 – Memória Mínima do Usuário: O usuário deve lembrar facilmente dos passos de operação, ou seja, o aprendizado do sistema deve passar por um processo hierárquico e incremental.

3 – Frustração mínima: o software deve evitar demora na execução de tarefas, e deve permitir que o usuário encontre a tarefa desejada rapidamente.

4 – Maximizar o uso de padrões e hábitos: o sistema deve possuir certas teclas para as mesmas funções. Além disso, deve apresentar um padrão quanto às posições das informações nas telas.

5 – Máxima tolerância para diferenças humanas: o sistema deve garantir o uso de métodos visuais e audíveis para chamar a atenção. Exemplo: sensores e alarmes. Desta forma, o sistema deve utilizar formas gráficas ou de desenho para auxiliar a comunicação visual.

6 – Notificação imediata de problemas: o sistema deve conseguir alertar ao usuário sobre um problema detectado.

7 – Controle máximo de tarefas pelo usuário: o usuário consegue controlar e gerenciar o seu sistema a partir de sua necessidade.

8 – Apoio máximo às tarefas: refere-se ao fato do operador do sistema não necessitar de outros recursos para conseguir executar uma determinada tarefa. O usuário não deve precisar de outros recursos para desempenhar a tarefa.

A Ergonomia de Software acaba sendo um item essencial para a construção de um sistema, uma vez que, gera uma melhor visualização estética, programas bem elaborados, redução dos custos de manutenção, eficiência, facilidades de aprendizado, etc.

5.2. Protocolos de Comunicação

Com a tendência atual de utilização de sistemas de informatização baseados em dispositivos de baixo custo, como microcomputadores, torna-se necessária a interligação desses dispositivos por meio de rede digital. Essa interligação permite o compartilhamento de recursos, como bancos de dados e a distribuição de tarefas e informações (ALBUQUERQUE & ALEXANDRIA, 2009).

Logo após o surgimento das redes de computadores, as soluções eram na maioria das vezes proprietárias, isto é, uma tecnologia só era suportada por seu fabricante; assim, um único fabricante era responsável pela construção de praticamente tudo na rede. Como era de se esperar, essa situação gerava problemas de custos, de expansibilidade dos sistemas e dificuldade para integrá-los (ROSÁRIO, 2005).

Portanto, um SCADA deve ter suporte aos mais variados tipos e fabricantes de equipamentos para que não haja nenhum dos problemas já citados.

Há uma pequena diferença em relação à comunicação tanto do ScadaBR como do Elipse SCADA, com equipamentos de um sistema automatizado. O ScadaBR já disponibiliza os protocolos de comunicação dos principais fabricantes, ou seja, é só escolher qual protocolo será usado no momento em que se cria os *Data Sources*, que é um “lugar” onde os dados são recebidos. No Elipse SCADA, a comunicação com os equipamentos de aquisição de dados e com outros computadores é feita executando-o através de *drivers* de E/S ou de rede



fornecidos pela Elipse Software, onde um documento é fornecido para cada um deles, com informações acerca de suas configurações.

Hoje em dia, os protocolos de comunicação entre os SCADAs e hardwares dos mais variados fabricantes não são um grande problema, vide que em instituições de ensino, os tipos de equipamentos proprietários disponíveis podem ser limitados.

5.3. Linguagens de Programação

No ScadaBR, quando se termina de configurar os protocolos de comunicação com os equipamentos e definem-se as variáveis de uma aplicação, consegue-se desenvolver telas interativas Web utilizando o próprio navegador. Além disso, o ScadaBR permite o desenvolvimento de aplicativos personalizados, em qualquer linguagem de programação atual (JAVA, C++, VB, PHP, JavaScript, MS EXcel entre outras), a partir do código-fonte disponibilizado ou de sua API “web-services” (ROCHA, 2010).

A grande vantagem de uma API deste tipo é que possibilita uma imediata integração entre os dados de processo (sistemas de chão-de-fábrica) e os dados gerenciais e estratégicos (sistemas corporativos) (ROCHA, 2010).

O Elipse SCADA detém uma linguagem de programação própria denominada Elipse Basic que apresenta semelhanças bastante similares às linguagens C, contudo com recursos de programação visuais parecidos com os do Visual Basic.

No Elipse, quando se declara variáveis ou funções, não precisa defini-los no início do Script, nele as variáveis de sistema já são pré-definidas.

A partir do que se foi definido acima, pode-se perceber que embora o software Elipse SCADA seja um software proprietário, ele possui uma linguagem de programação própria o que limita o operador do sistema a utilizá-la, enquanto o ScadaBR que é um software livre de código aberto, permite que usuário utilize qualquer linguagem de programação atual, acarretando em um leque de possibilidades na implementação dos sistemas. Porém, é importante ressaltar que a escolha do software deve partir do operador do sistema, uma vez que, ele determina qual a melhor opção para o desenvolvimento de sua aplicação. Desta forma, a linguagem de programação é um item essencial na lista de requisitos necessários para a implementação de um projeto utilizando softwares do tipo SCADA.

5.4. Banco de Dados

Um sistema de banco de dados é um sistema computadorizado cuja finalidade geral é armazenar informações e permitir que os usuários busquem e atualizem essas informações quando as solicitar (DATE, 2003).

Os sistemas de banco de dados oferecem diversos benefícios, dos quais um dos mais importantes é a independência de dados (física). A independência de dados pode ser definida como a imunidade de programas de aplicação a alterações no modo de armazenar fisicamente os dados e obter acesso a eles (DATE, 2003).

O Elipse SCADA utiliza o padrão para acesso a sistemas gerenciadores de bancos de dados ODBC do Windows, que efetua a manipulação do banco de dados enviando e recebendo dados. *Open Database Connectivity* (ODBC) é uma interface criada pela Microsoft, que oferece uma interface universal para acesso a diferentes bancos de dados incluindo Oracle, MS Access, MS Excel, MS Fox Pro, CA-Clipper, dBase, MySQL, Interbase, dentre outros.



O ODBC é uma interface de nível de chamada que permite que aplicações acessem dados em qualquer banco de dados para o qual há um driver ODBC. Essa relação de independência de um software de banco de dados específico faz com que tanto a manipulação, como a criação dos bancos de dados para qualquer aplicação utilizando o Elipse SCADA, gere uma facilidade para o usuário, que ao incorporar os arquivos contendo os dados necessários, os campos do Banco de Dados podem ser associados aos Objetos de Tela, bem como a Tags (variáveis de sistema).

O ScadaBR, atualmente, suporta dois sistemas gerenciadores de banco de dados: MySQL e Apache Derby, vindo configurado com este último por padrão por questões de facilidade para o usuário, pois o mesmo roda embarcado e não exige a instalação do SGBD ou a configuração de parâmetros como login e senha.

O Apache Derby é um banco de dados relacional implementado em Java. A alteração dos dados referentes ao banco utilizado é prevista para ser realizada durante a “construção” (*building*) do sistema, porém, caso seja necessário, pode-se alterá-las de outra maneira a partir do arquivo *ScadaBR.war* disponibilizado.

O banco de dados do Apache Derby é tão pequeno que é possível incorporá-lo facilmente em qualquer solução baseada em Java. Além de sua característica embarcada, ele suporta um quadro mais tradicional de cliente/servidor com o servidor de rede Derby, aumentando o poder do banco de dados permitindo conexão sobre o TCP/IP e suportar em rede o JDBC (*Java Database Connectivity*), ODBC, Perl e PHP.

Visando aplicações em instituições de ensino, onde não são realizadas aplicações muito complexas, que demandam bancos de dados extensos, o ScadaBR pode ser utilizado sem muitas restrições. Mas a escolha do software a ser usado, vai depender dos recursos que sua aplicação irá precisar.

5.5. Segurança

O ScadaBR é um software que garante o controle de acesso às aplicações a partir do sistema de identificação (login/senha). Além disso, o operador principal do sistema consegue determinar quais aplicações os usuários secundários podem ter acesso. Outro fator de extrema importância do ScadaBR é sua capacidade de enviar relatórios regulares via correio eletrônico para o administrador ou grupo de usuário pré-determinados, o que garante uma maior segurança quanto ao controle do sistema desenvolvido.

O Elipse SCADA permite controlar o acesso a uma aplicação através de uma lista de nomes, podendo atribuir uma senha a cada usuário e configurar níveis de segurança no seu sistema.

Pode-se perceber que o ScadaBR e o Elipse SCADA, apresentam praticamente os mesmos itens de segurança, o que os tornam excelentes softwares para o desenvolvimento de aplicações que envolvem uma quantidade maior de recursos.

5.6. Extensibilidade

A Extensibilidade representa o nível de facilidade com a qual se pode introduzir modificações futuras nos produtos de software.

No Elipse SCADA tem-se a possibilidade de se adicionar *Plug-Ins*, que são ferramentas adicionais que permitem a expansão dos recursos de um software, acrescentando novas funcionalidades ao mesmo, porém só é possível adicioná-los se forem produzidos pelo mesmo fabricante. Disponíveis para serem adquiridos separadamente, o ElipseWatcher e o ElipseWeb



são exemplos de *Plug-Ins* das funções básicas do SCADA. O primeiro permite a monitoração de sistemas através de recursos de captura, registro e transmissão digital de imagens em tempo real. Já o segundo, é um sistema para supervisão de processos através da internet utilizando qualquer navegador.

O ScadaBR possui uma arquitetura bastante amigável para se trabalhar com softwares de outros fabricantes. Pode ser executado em Windows, Linux e outros sistemas operacionais podem executar o software a partir de um servidor de aplicações qualquer, sendo o Apache Tomcat o padrão. Também é possível gerar relatórios de processos utilizando softwares de terceiros, como o *Pentaho* e o *iReport*. Há projetos dos colaboradores do Portal ScadaBR, que pretendem ampliar suas funções de software SCADA em módulos, porém está em andamento.

Assim, pode-se perceber que tanto o ScadaBR quanto o Elipse SCADA apresentam um conjunto de ferramentas de extensão bem ampla. Contudo, deve-se salientar que o Elipse SCADA, como foi dito acima, só permite a expansão de sua funcionalidade, a partir de recursos produzidos pelo mesmo fabricante, enquanto os recursos do ScadaBR estão sempre em um processo de evolução e adaptação com ferramentas de terceiros.

6. CONSIDERAÇÕES FINAIS

O presente trabalho teve como objetivo principal demonstrar a importância dos programas de supervisão, controle e aquisição de dados no âmbito interdisciplinar em engenharia, para uma melhor associação entre os conteúdos teóricos e práticos, visando à complementação do ensino na graduação. Desta forma, percebeu-se que um software livre como o ScadaBR pode ser utilizado em aplicações complexas e das mais diversas áreas, não deixando a desejar em capacidade quando comparado a um software proprietário, como o Elipse SCADA. Pode-se concluir que devido à dificuldade em que muitos estudantes dos cursos de engenharia têm em interligar os conhecimentos puramente teóricos àqueles que se limita somente ao laboratório, que o uso de softwares livres que, de alguma maneira, representam de forma visual e interativa dados de sistemas práticos reais, o aprendizado construtivista é fortalecido, instigando o conhecimento lógico, científico, tecnológico e criativo.

Agradecimentos

Os autores agradecem ao MEC/SESu/PET, FNDE, CAPES, FAPEMIG e CEFET-MG pelo apoio ao desenvolvimento deste trabalho.

REFERÊNCIAS BIBLIOGRÁFICAS

ALBUQUERQUE, P.U.B.; ALEXANDRIA, A.R.. Redes Industriais: Aplicações em sistemas digitais de controle distribuído. 2. ed. São Paulo: Ensino Profissional, 2009. 257 p, il.

Beremiz. **Beremiz**. Disponível em: <<http://www.beremiz.org/>> Acesso em: 21 mai. 2012.

BORGES, R. C. M. **Ergonomia de Software**. Disponível em: <<http://www.inf.ufrgs.br/~cabral/INF141.Cap.04.html>> Acesso em: 23 mai. 2012.



CERTI - ScadaBR. **Documento de Referência do ScadaBR.** Disponível em: <http://sites.google.com/a/certi.org.br/certi_scadabr/requisitos> Acesso em: 22 mai. 2012.

DATE, C.J. Introdução a Sistemas de bancos de dados. In: Visão geral do gerenciamento de banco de dados, Rio de Janeiro : Ed. Elsevier, 2003. p.[3]-27.

DISTEFANO, J.J. Sistemas de retroação e controle.1. ed. São Paulo: McGraw-Hill do Brasil, 1978.

Elipse Software HMI/SCADA Solutions. **Manual do Elipse SCADA.** Disponível em: <http://www.elipse.com.br/get/Elipse%20SCADA/v2.29/Build126/scadamanual_br.pdf> Acesso em: 16 mai. 2012.

IÓRIO, Luiz Carlos; UNICAMP, Faculdade de Engenharia Mecânica. Redes de comunicação em automação industrial: ênfase na solução tecnológica da plataforma Pipefa, 2002. Tese (Mestrado).

Likindoy The Free Scada. **¿Qué es Likindoy?.** Disponível em: <http://www.likindoy.org/es/Sobre-Likindoy/what_is_it> Acesso em: 18 mai. 2012.

Mango. **Mango, open source M2M.** Disponível em: <<http://mango.serotoninsoftware.com/>> Acesso em: 16 mai. 2012.

Microsoft Corporation. **MSDN Library: Open Database Connectivity (ODBC).** Disponível em: <<http://msdn.microsoft.com/pt-br/library/thzzea08.aspx>> Acesso em: 20 mai. 2012.

NISE, N. S. Engenharia de Sistemas de Controle.5. ed. Rio de Janeiro: LTC, 2011. 676 p, il.

Open Scada. **openSCADA.** Disponível em: <<http://openscada.org/>> Acesso em: 7 mai. 2012.

Portal do ScadaBR. **Manual do Software.** Disponível em: <<http://ufpr.dl.sourceforge.net/project/scadabr/Software/Docs/Manual%20ScadaBR.pdf>> Acesso em: 15 mai. 2012.

ROCHA, V. **Automação e Sensoamento Remoto utilizando Software Livre “SCADA”.** Disponível em: <<http://www.vivaolinux.com.br/artigo/Automacao-e-Sensoamento-Remoto-utilizando-Software-Livre-SCADA?pagina=3>> Acesso em: 14 mai. 2012.

ROSÁRIO, João Maurício. Princípios de Mecatrônica.1. ed. São Paulo: Prentice Hall, 2005. 348 p, il.

SOMMERVILLE, I. Software Engineering.6. ed. New York: Pearson Education, 2003. 285 p, il.

STRIKE, K.A.; POSNER, G.J. A revisionistic theory of conceptual change. ed. Albany, NY: SUNY, 1992. 113 p.147-176 il.



The Apache DB Projects. **Apache Derby.** Disponível em:
<<http://db.apache.org/derby/papers/DerbyTut/>> Acesso em: 22 mai. 2012.

Visual. **Visual web-enabled open source SCADA and HMI software for LINUX.**
Disponível em: <<http://visual.sourceforge.net/new/index.php>> Acesso em: 14 mai. 2012.

COMPARATIVE STUDY OF TOOLS FOR MONITORING AND CONTROL AND DATA ACQUISITION AND THE IMPORTANCE OF THESE FOR ENGINEERING EDUCATION

***Abstract:** This article presents a comparative study of two softwares for supervision, control and data acquisition, as well as the importance of these tools in engineering education. To perform this study two softwares that are widely used nationally in many areas of automation were chosen: Elipse SCADA and ScadaBR, to which parameters such as ergonomics software, protocols for communication with field devices, database systems, security of data and applications, among others, were applied. What is also noteworthy is that working with these tools, jointly with the construction of knowledge and an interdisciplinary practice, made possible to consolidate the learning of theoretical principles, bringing them through the simulated applications of real situations.*

***Key-words:** Control and automation Engineering, Mechatronics, SCADA, Constructivism, Ergonomics Software.*