**UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ**
**ESPECIALIZAÇÃO EM BANCO DE DADOS**

**JAIR PESSETI JUNIOR**

# EXPLORANDO ESQUEMAS DE DADOS SCADA PARA FINS DE AUTOMAÇÃO RESIDENCIAL

**MONOGRAFIA DE ESPECIALIZAÇÃO**

**PATO BRANCO**
**2017**

**JAIR PESSETI JUNIOR**

# EXPLORANDO ESQUEMAS DE DADOS SCADA PARA FINS DE AUTOMAÇÃO RESIDENCIAL

Trabalho de Conclusão de Curso, apresentado ao II Curso de Especialização em Banco de Dados, da Universidade Tecnológica Federal do Paraná, campus Pato Branco, como requisito parcial para obtenção do título de Especialista.

Orientador: Prof. Marcelo Teixeira.

**PATO BRANCO**
**2017**

# TERMO DE APROVAÇÃO

EXPLORANDO ESQUEMAS DE DADOS SCADA PARA FINS DE AUTOMAÇÃO RESIDENCIAL.

por

JAIR PESSETI JUNIOR

Este Trabalho de Conclusão de Curso foi apresentado em 23 fevereiro de 2017 como requisito parcial para a obtenção do título de Especialista em Banco de Dados. O(a) candidato(a) foi arguido(a) pela Banca Examinadora composta pelos professores abaixo assinados. Após deliberação, a Banca Examinadora considerou o trabalho aprovado.

_____
Marcelo Teixeira
Prof.(a) Orientador(a)

_____
Robison Cris Brito
Membro titular

_____
Vinicius Pegorini
Membro titular

**“O Termo de Aprovação assinado encontra-se na Coordenação do Curso”**

## RESUMO

PESSETI, Jair Pesseti Junior. Explorando esquemas de dados SCADA para fins de automação residencial. 2016. Monografia (II Curso de Especialização em Banco de Dados) - Universidade Tecnológica Federal do Paraná. Pato Branco, 2016.

Um tema que vem se destacado nos últimos anos no mercado de software é a automação residencial. Com a alta disponibilidade de tecnologia, ficou cada vez mais viável implementar políticas de controle e obter informações sobre uma determinada residência, em tempo real, na palma da mão. Uma ferramenta cuja utilização é fundamental em casos como esse é o SCADA, sistema por meio do qual é possível implementar o controle de componentes residenciais e monitorar o comportamento desses componentes remotamente, via browser. Sistemas SCADA são muito utilizados em diversos segmentos como indústrias, sistema de distribuição de energia, laboratórios e outros mais. Uma lacuna que se observa, no entanto, é relacionada à forma como os dados são armazenados e manipulados no SCADA. Em geral, esse tipo de aplicação despende uma preocupação maior com as políticas de controle e de monitoramento, ao passo que a manipulação de dados é, por vezes, apenas um complemento do software. É comum que a manipulação de dados em sistemas SCADA não passe da mera construção de relatórios e alarmes. Esse trabalho tem por objetivo explorar a estrutura do banco de dados dentro de um sistema SCADA. Será enfatizado como esse framework armazena dados e o que se pode fazer com eles usando os recursos do próprio SCADA. Isso será ilustrado por meio de um protótipo residencial. Posteriormente, será mostrado como o esquema de dados do SCADA pode ser melhor explorado com recursos avançados de programação, como triggers e procedures, e como esses recursos podem se integrar com o controle do processo para, por exemplo, acionar equipamentos automaticamente conforme certas condições pré-estabelecidas para a residência.



Palavras-chave: Scada. Automação Residencial. Banco de dados.

**ABSTRACT**

PESSETI, Jair Pesseti Junior. Exploiting database schemes in SCADA systems to home automation. 2017. Monography (II Specialization Course in Database) - Federal University of Technology - Parana. Pato Branco, 2017..

One topic that has attracted attention in recent years in the software market is residential automation. With the high availability of technology, it became increasingly feasible to implement control policies and obtain information about a particular residence, in real time, in the palm of your hands. An essential tool in cases is SCADA, a system through which it is possible to implement the control of residential components and monitor the behavior of these components remotely via the browser. SCADA systems are widely used in industry, in power distribution systems, academy, etc. One gap, however, is related to how data is stored and manipulated in SCADA. In general, this type of application spends a greater concern with control and monitoring policies, while data manipulation is sometimes only a complement to the software. It is common that data manipulation in SCADA systems is no more than the construction of reports and alarms. This work aims to explore the structure of the database within a SCADA system. It will be emphasized how this framework stores data and what can be done with them using SCADA's own resources. This will be illustrated by means of a residential prototype. Later, it will be shown how the SCADA data schema can be better exploited with advanced programming features such as triggers and procedures, and how these features can be integrated with process control to, for example, trigger equipment automatically according to certain preconditions established for the residence.

Palavras-chave: SCADA. Home Automation. Databases.

**LISTA DE FIGURAS**

## LISTA DE QUADROS

**SUMÁRIO**

# 1 INTRODUÇÃO

Este capítulo apresenta as considerações inicias, com uma visão geral do trabalho, os objetivos, a justificativa e a organização do texto.

## 1.1 CONSIDERAÇÕES INICIAIS

A automação Residencial é um segmento que vem se destacando cada vez mais com o avanço da tecnologia. Com a jornada de trabalho de em média 8 horas por dia, sem contar as horas extras, o tempo gasto com os filhos, o trânsito e outros afazeres fora de casa, as famílias acabam não despendendo o tempo necessário para tarefas simples como fechar as janelas em dias em que há a possibilidade de chuva, ou ativar o alarme ao sair de casa.

Com o intuito de tornar a vida mais cômoda, prática e segura surgiu a ideia da automação residencial, fundamentada, sobretudo na utilização de *softwares* especializados, como por exemplo, os Sistemas SCADA, os quais são objeto desse estudo.

Um Sistema SCADA**,** *Supervisory Control And Data Acquisition*, o que significa Controle Supervisório e Aquisição de Dados, pode ser utilizado em muitos segmentos, como a automação residencial, que já foi citada, industrial, em redes de energia, saneamento, entre outros. O sistema consiste basicamente em uma página *Web* onde se pode montar pontos para coleta de dados, tanto virtualmente, gerando dados aleatórios, quanto fisicamente, através de *hardwares* específicos para coleta das informações desejadas.

Existem opções de criar relatórios de informações em tempo real onde o usuário pode montá-los como mais agradar dependendo do resultado que se quer alcançar. No momento da instalação do *software*, o usuário escolhe, dentre diversas opções de configuração, as opções disponíveis para o armazenamento de dados. No caso da marca específica de sistema SCADA que foi utilizada nesse trabalho, o ScadaBR, tem-se duas opções de banco de dados, Derby e MySQL, Nesse estudo foi optado pela utilização do MySQL.

A principal limitação do Sistema SCADA e também motivo desse estudo é a falta de integração das funcionalidades do sistema com o banco, a carência de informações referentes ao banco de dados e sua, estrutura e como as informações são salvas. A falta de conhecimento sobre as ações que poderiam ser realizadas pelo banco de dados tornam ineficaz a manipulação dos dados coletados no sistema, já que a interface *Web* permite limitadamente que instruções em SQL sejam realizadas.

## 1.2 OBJETIVOS

Nessa seção são apresentados os principais objetivos desse estudo de caso assim como cada objetivo específico que permeiam a sua realização.

### 1.2.1 OBJETIVOS GERAIS

Implementar um protótipo residencial no sistema SCADA para coleta e armazenamento de dados residenciais gerados virtualmente. Explorar a base de dados para a implementação de ações automatizadas sobre a residência, usando diretamente o banco de dados, retirando essa implementação do aplicativo.

### 1.2.2 OBJETIVOS ESPECÍFICOS

- Verificar a estrutura do banco de dados integrado ao scada e gerar a modelagem para conhecimento estrutural.
- Implementar o protótipo de uma Automação Residencial para fins de estudo, no qual as informações serão coletadas através de um *Data Source* virtual, ou seja, gerando dados por simulação.
- Verificar como as informações são salvas nas tabelas e como se comportam quando determinada ação acontece.
- Criar ações diretamente no banco de dados para fim de controle onde essas ações fiquem visíveis na interface do sistema SCADA.

## 1.3 JUSTIFICATIVA

O sistema SCADA é um excelente *software* para controles gerenciais, e o estudo em cima da base de dados é justificado pela importância de se ter um conhecimento mais profundo sobre o assunto para que ações que não são possíveis de serem realizadas pela interface do sistema SCADA possam ser feitas pelo banco de dados.

Quando se tem um conhecimento mais aprofundado sobre a forma como o banco de dados é utilizado e como as suas informações são salvas, há tendência de que isso reflita positivamente ao sistema como um todo.

# 2 REFERENCIAL TEÓRICO

Este capítulo apresenta o referencial teórico utilizado para fundamentar o estudo proposto. O texto esta focado em definir e exemplificar o uso do Sistema ScadaBR com fim de automação residencial, utilizando o Banco de Dados MySQL.

## 2.1 AUTOMAÇÃO RESIDENCIAL

Segundo MURATORI; DAL BÓ, (2015), a Automação residencial surgiu entre as décadas de 1970 e 1980 nos Estados Unidos, com intuito de facilitar a vida dos moradores, permitindo o controle centralizado e integrado de diferentes tipos de equipamentos em uma residência. Hoje em dia, a automação residencial tem ganhado uma dimensão imensurável.

No contexto dos sistemas cyber-físicos, estima-se que seja possível romper a barreira entre os mundos real e virtual, fazendo com que componentes de uma residência interajam de igual para igual com as pessoas.

Automação Residencial ou também conhecida como “domótica” que é originado da junção das palavras *Domus*, que em latim significa casa, e robótica, que representa uma tecnologia capaz de controlar todos os ambientes de uma residência através de um só equipamento, incluindo temperatura, luminosidade, som, segurança, dentre outros, ou seja, automação residencial (BOLZANI, 2004; FERREIRA, 2008; SGARBI , 2007).

A ideia por traz da automação residencial é possibilitar o controle de equipamentos elétricos e eletrônicos sem a necessidade de intervenção humana, por meio de algum sistema de controle, utilizando sensores para as devidas decisões, disparando intenções ou mudando o ambiente (BOLZANI, 2010). Os comandos do sistema de automação são recebidos por dispositivos eletromecânicos chamados de atuadores, que ativam os equipamentos automatizados. Portanto, os atuadores são os módulos de acionamento ligados entre a rede elétrica e os equipamentos (ALMEIDA, 2009).

Para corresponder às exigências, a domótica faz uso de vários equipamentos distribuídos pela residência de acordo com as necessidades dos moradores. Estes equipamentos podem ser divididos em três principais grupos (TAKIUCHI et al, 2004):

- Atuadores: controlam os aparelhos da residência como, por exemplo, luz e ventilador;
- Sensores: capturam informações do ambiente como, por exemplo, luminosidade, umidade e presença;

- Controladores: são responsáveis pela administração dos atuadores e sensores, ou seja, coordenam todos os aparelhos e equipamentos da residência que fazem parte da automação.

Com a diminuição dos custos de equipamentos como computadores pessoais e componentes eletrônicos utilizados para a fabricação de *hardwares*, bem como com o advento da Internet e com o avanço tecnológico utilizado para o desenvolvimento de *softwares*, tornou-se inevitável o surgimento da automação residencial.

Inicialmente, a automação residencial foi uma adaptação da automação industrial para residências, mas devido às visíveis diferenças entre um ambiente residencial e um industrial, veio ela própria a tornar-se uma nova linha de pesquisa e uma considerável fonte de investimentos (BOLZANI, 2004).

## 2.2 SCADABR

Sistemas SCADA servem como interface entre o operador e processos dos mais variados tipos, como máquinas industriais, controladores automáticos e sensores dos mais variados tipos. Com sistemas SCADA são construídos desde aplicativos simples de sensoriamento e automação, até os famosos "Painéis de Controle" em empresas de geração e distribuição de energia elétrica, centrais de controle de tráfego e assim por diante.

Um SCADA típico deve oferecer *drivers* de comunicação com equipamentos, um sistema para registro contínuo de dados ("*datalogger*") e uma interface gráfica para usuário, conhecida como "IHM" ou Interface Homem-Máquina. Na IHM são disponibilizados elementos gráficos como botões, ícones e *displays*, representando o processo real que está sendo monitorado ou controlado. Entre algumas das funções mais utilizadas em sistemas SCADA estão:

- Geração de gráficos e relatórios com o histórico do processo;
- Detecção de alarmes e registro de eventos em sistemas automatizados;
- Controle de processos incluindo envio remoto de parâmetros e *set-points*, acionamento e comando de equipamentos;
- Uso de linguagens de *script* para desenvolvimento de lógicas de automação;

Devido ao aumento significativo da quantidade de estudantes que buscam formação profissional, ocorre também um crescimento da necessidade do aumento da capacidade dos

laboratórios e dos recursos didáticos, além dos recursos físicos para satisfazer o aprendizado de todos os formandos (JUCÁ, 2006).

Nesse sentido, nasceu o ScadaBR, um *software open-source* implementado em java, com um servidor *web* onde à vários segmentos que podem ser controlados e monitorados pelo *software*, dentre eles automação residencial, a qual esta sendo abordada neste estudo. Ele pode ser usado tanto no contexto didático/educativo, pois possui recursos que podem ser utilizados em contextos de ensino-aprendizagem, quanto no contexto industrial, uma vez que é empregado amplamente na indústria, em casos reais de controle e monitoramento de processos.

O *software* para uso comercial ScadaBR se enquadra em *software* de simulação, pois apresenta na tela, a modelagem de um sistema ou situação real, por meio de gráficos e imagens animadas. É um *software* que oferece um ambiente exploratório onde o discente/usuário pode tomar decisões e em seguida comprovar as consequências.

O ensinamento de temas complexos ou impossíveis de se observar é facilmente analisado por este *software*. Além disso, o *software* ScadaBR acaba agindo de forma construtiva no desenvolvimento de habilidades lógicas, matemáticas e de resolução de problemas (SANCHO, 1998).

Segundo estudo feito e comparado com a grade curricular do curso de Engenharia de Controle e Automação da instituição CEFET-MG Campus Leopoldina pode-se aplicar o *software* ScadaBR nas disciplinas:

- Contexto Social e Profissional do Engenheiro de Controle e Automação: abordando a utilização do ScadaBR nos cenários da engenharia de controle e automação no Brasil e no mundo;
- Introdução à Experimentação e ao Desenvolvimento de Protótipos e Projetos: através de orientação à concepção, planejamento e construção de projetos experimentais por meio do ScadaBR;
- Química: através de simulações e controle de fenômenos e processos químicos;
- Físicas I, II e III: através de simulações dos fenômenos básicos da Mecânica, da Termologia e da Eletricidade;
- Materiais Elétricos: verificando através da simulação as características fundamentais de funcionamento dos materiais;
- Sistemas Digitais: simulação e análise de projetos de sistemas digitais;
- Controle Automático: simulação do comportamento de sistemas de controle;

- Informática Aplicada: simulação, por exemplo, do funcionamento de CLP's;
- Metrologia e Sensores: criação de sistemas industriais nos quais se encontram diversos tipos de sensores;
- Instrumentação e Controle: simulação de aplicações dos sistemas de aquisição e processamento automático de dados.

Os *softwares* educativos podem estimular o desenvolvimento do raciocínio lógico e a autonomia do indivíduo, à medida que se podem levantar hipóteses, fazer interferências e obter conclusões a partir dos resultados apresentados (BORGES, 1999)

Conforme Garbrecht (2012), os desenvolvimentos de aplicações de supervisão SCADA oferecem até 80% de economia no custo de engenharia.

Utilizando integração com o Google Maps, o ScadaBR é a base para supervisão das unidades da CASAN na região da Grande Florianópolis. Implementado pela equipe técnica da CASAN/GPO/DIPAE em parceria com a Systomatus, o Sistema Remoto de Monitoramento tornou-se uma ferramenta essencial para supervisão e controle das estações elevatórias e de tratamento.

A visibilidade dos pontos monitorados pelo sistema oferece ao centro de operação a possibilidade de agir rapidamente em casos de falhas e prevenir ocorrências com o crescimento da demanda. Velocidade em resposta aos clientes da CASAN garante qualidade e assiduidade no fornecimento de água e assegura ao meio ambiente o mínimo impacto (SCADABR).

Abaixo são os principais elementos do ScadaBr para desenvolvimento de projetos:

### 2.2.1 *Data Source*

*Data sources* (fontes de dados) são partes fundamentais para a operação desta aplicação. Um *data source* é um "lugar" de onde os dados são recebidos. Virtualmente, qualquer coisa pode ser um *data source*, desde que o protocolo de comunicação seja suportado pela aplicação. Alguns exemplos:

• Uma rede *Modbus* acessível por RS232, RS485, TCP/IP ou UDP/IP, pode ser monitorada pode meio de um *data source Modbus* que irá "*pollar*" (*poll*) a rede em um intervalo definido.

• Equipamentos ou aplicações que podem enviar dados sobre HTTP, podem ser monitorados por meio de um data source HTTP receiver que irá escutar conexões recebidas e enviar os dados aos pontos apropriados.

• Para *hardware* que suporta SNMP, pode ser configurado um *data source* SNMP. Valores poderão ser "pollados" em intervalos definidos, ou *traps* podem ser recebidos para *report-on-exception*.

• Dados podem ser lidos e atualizados em uma base de dados SQL externa ao sistema.

• Dados podem ser gerados randomicamente ou preditivamente usando um *data source* Virtual, que será a forma utilizada para o desenvolvimento deste protótipo. De maneira geral, independente do tipo do *data source*, os valores de dados recebidos ou coletados por um *data source* são armazenados em *data points*.

• É possível se ter *Data Sources* chamados de *Data Sources Meta* onde seus *Data Points* tem a capacidade de utilizar de informações geradas/coletadas por outros *Data Points* associados a outros *Data Sources* ao invés de obter sua informação de uma fonte externa, e permite manipulação arbitrárias pelo usuário. (MANUAL DO SOFTWARE - SCADABR)

### 2.2.2 *Data Point*

Um *data point* é uma coleção de valores históricos associados. Por exemplo, um ponto particular pode ser uma leitura de temperatura de um quarto, enquanto outro ponto poderia ser a leitura de umidade do mesmo quarto. Pontos também podem ser valores de controle, como um indicador para ligar ou desligar um equipamento.

Inicialmente existe o conceito de um *point locator*. *Locators* são usados por *data sources* para determinar como "achar" os dados para o ponto particular. Por exemplo, um *data source SQL* tem atributos incluindo onde achar a instância da base de dados; *point locators* para o *data source* indicam o nome da tabela e dos campos onde podem ser achados valores específicos. A separação lógica de *data source* e de *data point* dependem do protocolo de comunicação em questão.

Atributos de *data points* também podem determinar muitos outros aspectos do ponto, como seu nome, como deve ser registrado (todos os dados, apenas mudanças no valor, ou nenhum), por quanto tempo manter os dados, como formatar os dados para exibição e como traçar um gráfico com os valores.

Também é possível configurar data points com detectores de valor, que são usados para detectar condições de interesse nos valores dos pontos, como por exemplo, se o valor esteve muito alto por muito tempo, se é muito baixo, se muda com frequência, se não muda, etc.

Pontos podem ser arranjados em uma hierarquia, ou árvore, para simplificar sua gerência e exibição usando a funcionalidade de Hierarquia, disponíveis no ScadaBr. (MANUAL DO SOFTWARE - SCADABR)

### 2.2.3 *Point Links*

*Point links* são utilizados para atualizar o valor de pontos baseados no valor de outros pontos, mantendo os dois pontos em sincronia de acordo com alguma relação definida pelo usuário. De maneira mais direta, os *point links* podem ser utilizados para ler valores em um sistema (de um ou mais *data sources*) e imediatamente definir valores em outro sistema (para um ou mais *data sources*). (AJUDA SCADABR)

### 2.2.4 *Scripting*

*Scripts* são procedimentos que automatizam a execução de tarefas, permitindo o controle dos *data sources* e *data points*. Os *scripts* não possuem valor de retorno, no entanto permitem configurar o valor de um *data point* do sistema, se o mesmo for configurável. Para fazer a utilização dos *Scripts* deve se conhecer da Linguagem de programação JavaScript.(AJUDA SCADABR)

### 2.2.5 Eventos

Um evento é a ocorrência de uma condição definida no sistema. Existem tanto eventos definidos pelo sistema como definidos pelo usuário. Eventos definidos pelo sistema incluem erros de operação de *data sources*, *logins* de usuários, e inicialização e parada do sistema.

Eventos definidos pelo usuário incluem detectores de valor, eventos agendados, e eventos compostos que detectam condições sobre pontos múltiplos usando argumentos lógicos. Há também os "eventos auditados" que ocorrem quando usuários fazem alterações (adições, modificações e remoções) que afetam objetos em tempo de execução, incluindo *data sources*, *data points*, detectores de valor, eventos agendados, eventos compostos e tratadores de eventos.

Uma vez que um evento foi detectado, ele é manipulado por tratadores. Um tratador de eventos é um comportamento definido pelo usuário que deve ser executado quando um evento particular ocorre, como envio de email ou "setar" o valor a um ponto configurável. (MANUAL DO SOFTWARE - SCADABR)

### 2.2.6 Relatórios

O ScadaBR possui um gerador de relatórios próprio, além de ser compatível com os principais geradores de relatórios customizados como por exemplo o Pentaho.

Para geração de relatórios é possível definir os *Data Points* que serão incluídos no relatório, como por exemplo, um *Data Point* que seja um armazenador de temperatura. Então o relatório mostraria o histórico de temperatura que foi registrado adicionado filtros como período de datas, etc. (MANUAL DO SOFTWARE - SCADABR)

## 2.3 MYSQL

O MySQL é um servidor e gerenciador de banco de dados relacional que utiliza a linguagem SQL (MYSQL, 2011). Ele é de licença dupla (*software* livre e paga), projetado inicialmente para trabalhar com aplicações de pequeno e médio portes, mas atualmente atende também aplicações de grande porte.

As principais características incorporadas na versão 5 do MySQL (MILANI, 2007) são:

a) Visões - são consultas pré-programadas ao banco de dados que permitem unir duas ou mais tabelas e retornar uma única tabela como resultado. Além disso, podem ser utilizadas para filtrar informações, exibindo somente os dados específicos de uma determinada categoria de uma ou mais colunas da tabela. Com o uso de visões, operações frequentes com uniões de tabelas podem ser centralizadas. É possível também utilizá-las para controle de acesso, permitindo que determinados usuários acessem dados de uma visão, mas não as tabelas utilizadas para compor a visão, restringindo acesso a informações.

b) Cursores - cursores possibilitam a navegação em conjuntos de resultados. De forma simples, é possível navegar pelos registros de uma tabela a partir de laços de repetição, permitindo realizar operações necessárias e transações à parte para cada linha da tabela.

c) *Information Schema* - são tabelas responsáveis apenas pela organização dos recursos do banco de dados, conhecidos como dicionário de dados, ou metadados. Desta forma, é possível realizar consultas sobre a estrutura do banco de dados por meio dessas tabelas.

d) Transações distribuídas XA - são uma espécie de extensão da ACID (Atomicidade, Consistência, Isolamento, Durabilidade) que fornece a possibilidade de gerenciamento dessas transações realizadas com a união de múltiplos bancos de dados (transações distribuídas) para a execução de uma mesma transação. Por exemplo, em determinadas situações pode surgir a

necessidade de integração de duas bases de dados distintas, mas que em algum momento necessitem uma da outra para realizar uma operação.

e) Integridade referencial - os relacionamentos entre tabelas distintas são gerenciados pelo banco de dados quando de inclusão, alteração e exclusão. Esse recurso visa manter as relações existentes no banco de dados confiáveis.

f) Clusterização - a clusterização é baseada na integração e sincronismo de dois ou mais servidores para dividirem a demanda de execuções entre si. Além da sincronização de um *cluster*, é possível especificar um balanceador de cargas. Desta forma, esse recurso gerenciará o redirecionamento de servidores secundários no caso de parar o funcionamento e balanceará as consultas recebidas pelo *cluster*, distribuindo-as pelos servidores de acordo com sua disponibilidade.

## 3 MATERIAIS E MÉTODOS

As tecnologias utilizadas implementação do protótipo foram:

a) ScadaBr para implementação de Protótipo;

b) MySQL para o banco de dados;

Primeiramente instalado MySQL e criado instância de banco de dados chamada scadabr, pois para a instalação do ScadaBR utilizando banco de dados MySQL é necessário configurar a instância primeiro, a partir disso a própria instalação do ScabaBR cria as tabelas necessárias para utilização do *software*.

Depois de feita instalação do ScadaBR foram criados cinco *Data Sources* e dezesseis *Data Points* os quais serão explicados a criação e configuração no capítulo 4 – Desenvolvimento.

## 4 RESULTADOS

Essa seção mostra um exemplo de aplicação Scada, implementado usando o ScadaBR, para o controle e o monitoramento de uma residência, considerando a integração desse sistema com um banco de dados MySQL.

Considerando uma residência com dois quartos, cozinha, sala, banheiro, como mostrado na Figura 1. Foi criada a automação com as seguintes ações para facilitar e dar conforto e segurança ao usuário:

- Ligar/Desligar todas as lâmpadas através de um botão correspondendo à determinada lâmpada;
- Abrir/Fechar todas as janelas através de um botão correspondendo à determinada janela;
- Fechar todas as janelas e desligar todas as lâmpadas em um único botão simulando que o usuário estivesse saindo de casa;
- Fechar as janelas caso comece a chover;
- Ligar automaticamente um Ventilador/Ar Condicionado caso a temperatura ultrapasse 25°;
- Ligar equipamento de alimentação de Pets em determinado horário configurado;
- Enviar email ao usuário caso a temperatura da residência ultrapasse 39°.

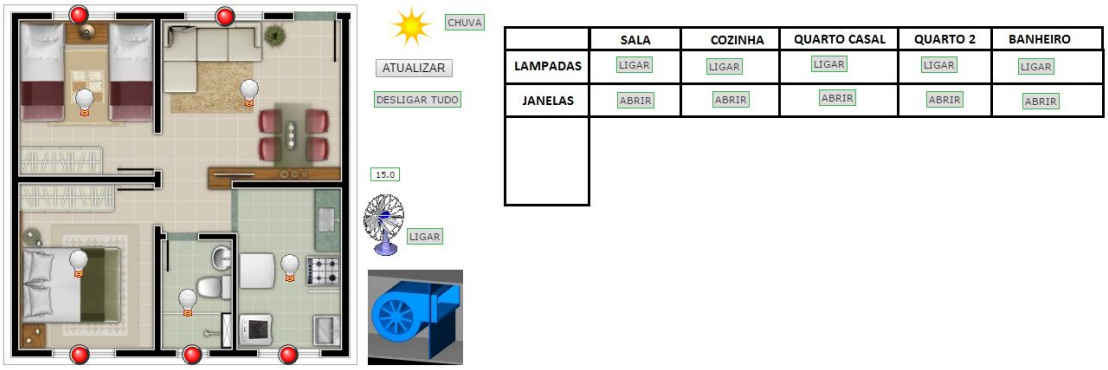


**Figura 1 – Interface gráfica para controle da residência**

Abaixo será detalhado o processo de desenvolvimento do protótipo.

## 4.1 *DATA SOURCES*

Para a coleta/geração de dados foram utilizados quatro *Data Sources* Virtuais (DS_LAMPADAS, DS_EXTERNO, DS_INTERNO e DS_TEMP) e um *Data Source Meta* (DS_META), como mostrado na Figura 2. Os *Data Sources* Virtuais irão gerar dados aleatórios simulando as informações coletadas em uma residência, já o *Data Source Meta* tem a capacidade de combinar pontos existentes em novos, ao invés de obter sua informação de uma fonte externa, utiliza valores de outros pontos e permite manipulação de maneiras arbitrárias pelo usuário.


| Nome | Tipo | Conexão | Status | |
|---|---|---|---|---|
| DS_EXTERNO | Data Source Virtual | 3 segundo(s) | | |
| DS_INTERNO | Data Source Virtual | 1 segundo(s) | | |
| DS_LAMPADAS | Data Source Virtual | 3 segundo(s) | | |
| DS_META | Data Source Meta | | | |
| DS_TEMP | Data Source Virtual | 30 segundo(s) | | |

**Figura 2 - *Data Sources* criados para desenvolvimento do protótipo**

## 4.2 *DATA POINTS*

Associados a cada *Data Source* existem *Data Points* que fazem a coleta de dados. Os *Data Points* foram divididos em seus determinados *Data Sources* para ficarem mais organizados e também por configurações específicas como o tempo de busca de dados, por exemplo, DS_EXTERNO buscará informações a cada 3 Segundos enquanto DS_TEMP a cada 30 segundos.

Abaixo será descrito todos os *Data Points* associados a cada *Data Source*, como mostrado nas Figuras 3, 4, 5, 6 e 7 .

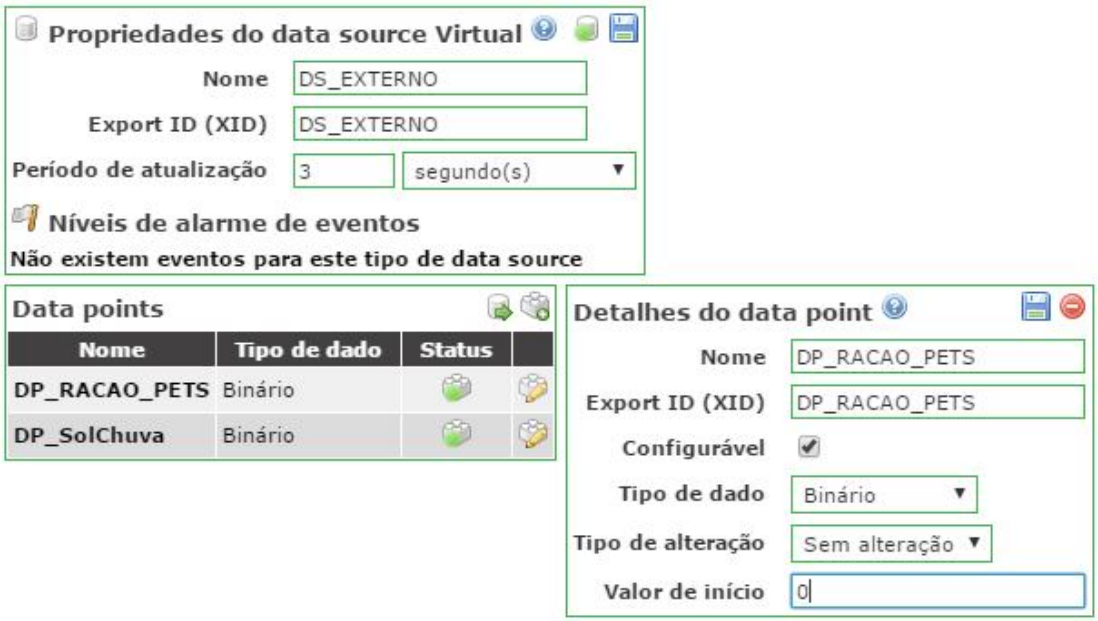


**Figura 3 – *Data Source* DS_EXTERNO e seus *Data Points***

Foram criados dois *Data Poitns* do tipo Binário ao *Data Source* DS_EXTERNO:

- DP_RACAO_PETS: quando o valor coletado for igual a Um (1) é por que a máquina que alimentará os pets está ligada e Zero (0) Desligada;
- DP_SolChuva: quando o valor coletado for igual Um (1) é por que está chovendo e Zero (0) o clima está ensolarado;

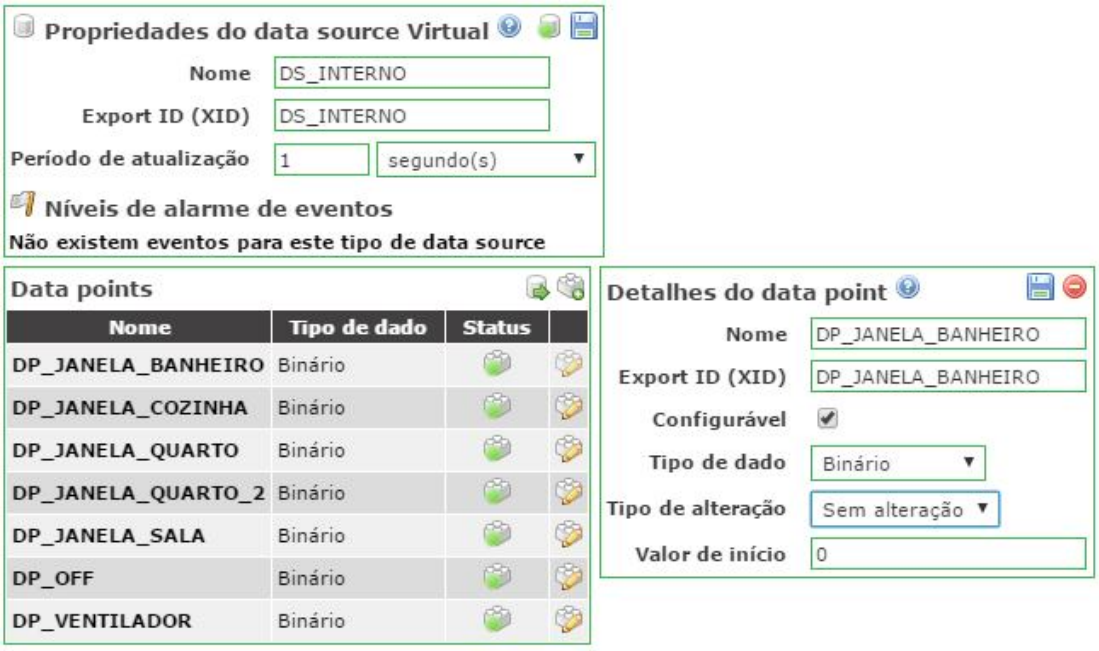


**Figura 4 – *Data Source* DS_INTERNO e seus *Data Points***

Foram criados sete *Data Points*, todos do tipo binário, onde definimos o Tipo de Alteração como “Sem alteração”, ou seja, ele não trocará de valor automaticamente, apenas quando o usuário informar seu valor:

- Todos os que estiverem representando uma Janela (DP_JANELA_BANHEIRO; DP_JANELA_COZINHA; DP_JANELA_QUARTO; DP_JANELA_QUARTO_2; DP_JANELA_SALA) serão configurados da seguinte forma: quando o valor coletado for igual a Zero (0) é por que a janela está fechada e igual a Um (1) a janela está aberta;
- DP_OFF: Esse *Data Point* tem a função de quando seu valor for alterado não imprta se pra Zero (0) ou Um (1), é por que o usuário está saindo de casa, então ele gerará um evento que fechará todas as janelas e apagará as lâmpadas;
- DP_VENTILADOR: : quando o valor coletado for igual a Zero (0) é por que o ventilador está desligado e igual a Um (1) o mesmo está ligado;


**Figura 5 – *Data Source* DS_LAMPADAS e seus *Data Points***

Foram criados cinco *Data Points* todos do tipo binário, onde definimos o Tipo de Alterção como “Sem alteração”, assim como no anteiror (DS_INTERNO). Quando o valor coletado for igual a Zero (0) é por que a lâmpada está desligada e igual a Um (1) a lâmpada está ligada.

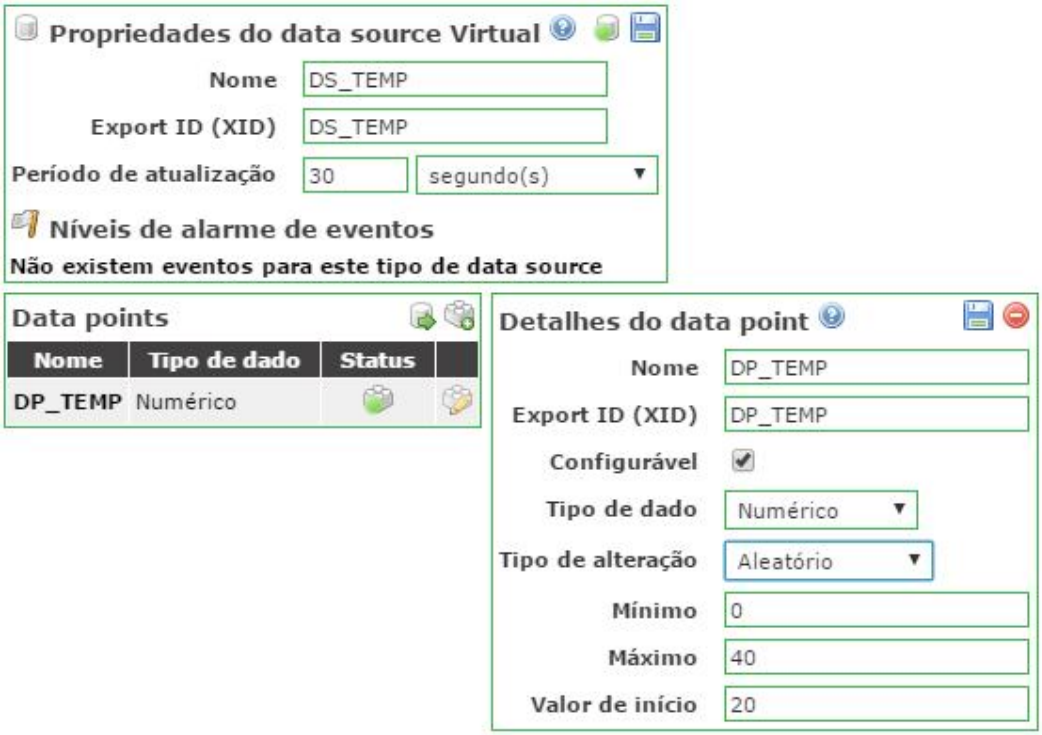


**Figura 6 – *Data Source* DS_TEMP e seu *Data Point***

Neste *Data Source* existe apenas um *Data Point*, esse do tipo Numérico, onde será feita a geração virtual da temperatura do ambiente variando de 0°C a 40°C a cada 30 segundos, onde definimos o valor de início para 20°C.

Como já explicado um *Data Source Meta* é diferente do Virtual. Neste o usuário tem configurações sobre determinadas ações e assim alarmes de eventos podem ser disparados conforme a programação escolhida pelo usuário.

Nesse *Data Source* também foi criado um *Data Point* do tipo Binário, mas quando é criado dentro de um *Data Source Meta* as configurações são diferentes, o usuário pode criar um *Script* de Execução na Linguem JavaScript a partir do valor de um outro *Data Point* ligado à outro *Data Source*.

Abaixo é mostrado um exemplo utilizado no protótipo onde o DP_VENTILADOR_TEMP servirá para monitorar os valores do Data Point DP_TEMP do Data Source DS_TEMP. O *Script* criado foi para que quando o valor do DP_TEMP for maior que 25°C retorne um valor para o DP_VENTILADOR_TEMP igual a *True* caso contrário *False*.

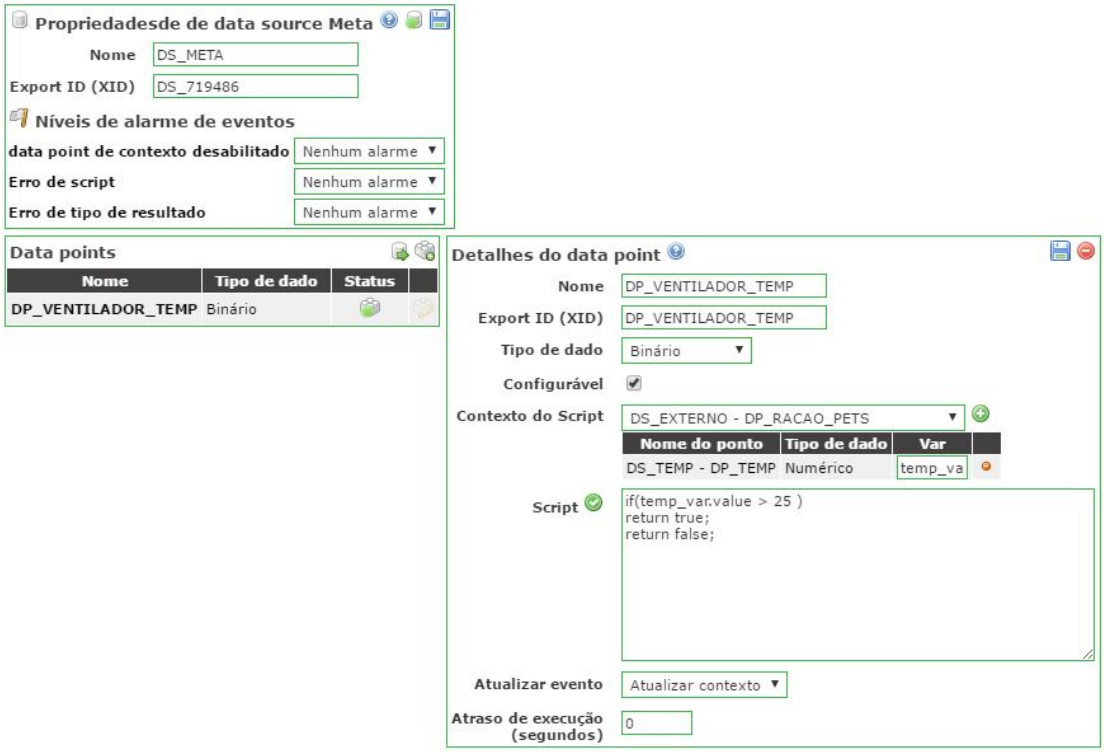


**Figura 7 – *Data Source* DS_META e seu *Data Point***

## 4.3 *POINT LINKS*

Como foi explicado *Point Links* são utilizados para "Ligar" um *Data Point* a outro, ou seja, quando determinada ação acontecer no *Data Point* origem pode-se utilizar o mesmo para criar uma ação no *Data Point* destino. Neste projeto utilizamos um *Data Point Link* fazendo a ligação do *Data Point* DP_VENTILADOR_TEMP que está no *Data Source Meta* chamado DS_META com o DP_VENTILADOR do *Data Source* DS_INTERNO.

Esse *Point Link* será utilizando para acionar o ventilador conforme o valor de retorno do *Data Point* origem, como mostrado na Figura 8.


**Figura 8 – *Point Link* entre os *Data Points* DP_VENTILADOR_TEMP e DP_VENTILADOR**

### 4.4 *SCRIPTING*

Utilizando-se da linguagem de programação Java Script foi criado um *Scripting* para fazer a atualização de dois *Data Sources*, quando são inseridos registros diretamente no banco de dados é preciso fechar e abrir a conexão para que eles recebam as informações inseridas no banco de dados.

Abaixo exemplo de como desativar e ativar um *Data Source* utilizando os comandos *disableDataSource* para desativar e *enableDataSource* para ativar. Para utilizar esses comandos é necessário definir um "alias/variável" como demonstrado na Figura 9.


**Figura 9 – *Script* para desabilitar e habilitar DS_INTERNO e DS_LAMPADAS**

### 4.5 EVENTOS ASSOCIADOS À DATA POINTS, EVENTOS AGENDADOS E TRATADORES DE EVENTOS

É possível criar dois tipos de Eventos, os associados à *Data Points* e os Agendados, os eventos associados à *Data Points* são utilizados quando o usuário deseja receber um alerta na tela quando determinada informação seja coletada pelo *Data Point*.

Eventos Agendados são utilizados quando usuário deseja que determinada ação aconteça em determinada horário, diariamente, semanalmente, mensalmente e etc., como por exemplo, inserir uma informação em um *Data Point.*

Os tratadores de eventos servem para monitorar os eventos dos *Data Points* e Eventos Agendados e assim poder fazer determinada ação.

Abaixo temos três Eventos associados a *Data Points* criados no protótipo:

- Evento associado ao *Data Point* DP_SolChuva, onde controlamos a mudança de valor e o tempo que permanecer no valor coletado, então conforme ilustrado na Figura 10 caso o valor coletado for igual a UM e permanecer por 3 segundo ou mais, o sistema disparará um evento chamando CHUVA que é a mensagem informada no campo ALIAS, neste caso definimos o nível do alarme como “Nenhum Alarme”, pois não é necessário que seja apresentado nada em tela para o usuário, esse evento será utilizado apenas no banco de dados para fechar todas as janelas, onde será explicado na seção de Banco de Dados.


**Figura 10 – Evento CHUVA quando o *Data Point* DP_SolChuva obter um valor igual a um(1)**

- Este segundo evento associoado ao *Data Point* DS_OFF também será controlado a mudança de valor, mas diferente do anterior, como mostrado na Figura 11, neste não importa o valor que for coletado, ou seja caso haja alteração de valor é sinal que o usuario esta saindo de casa, neste também o evento gerado será utilizado no banco de dados para fechar todas as janelas e desligar todas as lâmpadas via banco de dados.


**Figura 11 – Evento CASA_OFF quando o *Data Point* DP_OFF trocar de valor**

- Neste evento associonado ao *Data Point* DS_TEMP será utilizado para disparar um evento caso a temperatura coletada for a cima de 39°C, onde foi definido o tipo como Limite Superiror e sua duração de um segundo, ou seja caso o valor ultrapasse o valor de 39 por mais de um segundo disparará o evento chamado Pegando Fogo, como mostrado na Figura 12. Esse evento vamos usar atraves dos Tratadores de Eventos para enviar um email para usuario informando o ocorrido.


**Figura 12 – Evento Pegando Fogo quando o *Data Point* DP_TEMP obter um valor acima de 39°**

Toda vez que um evento é disparado cai nas rotinas de tratadores para verificar se existe algo configurado para o ocorrido, então utilizando o evento a cima, para enviar um e-mail ao usuário é apenas necessário criar ao Evento “Pegando Fogo” uma configuração do tipo e-mail, definir a mensagem e o usuário ao ser enviado, então toda vez que o evento for disparado automaticamente será enviado um e-mail através dessa configuração, como mostrado na Figura 13.


**Figura 13 – Configuração para enviar email quando ocorrer o evento Pegando Fogo**

A Figura 14 ilustra o email recebido pelo usuário ao disparar o evento.


**Figura 14 - Email enviado ao disparar evento**

Para exemplificar os Eventos agendados foi criado um que terá a função de acionar o dispositivo de alimentar os Animais através de uma configuração por horário, onde foi definido que será executado diariamente as 12h00min e terá fim 30 segundos após, como mostrado na Figura 15.

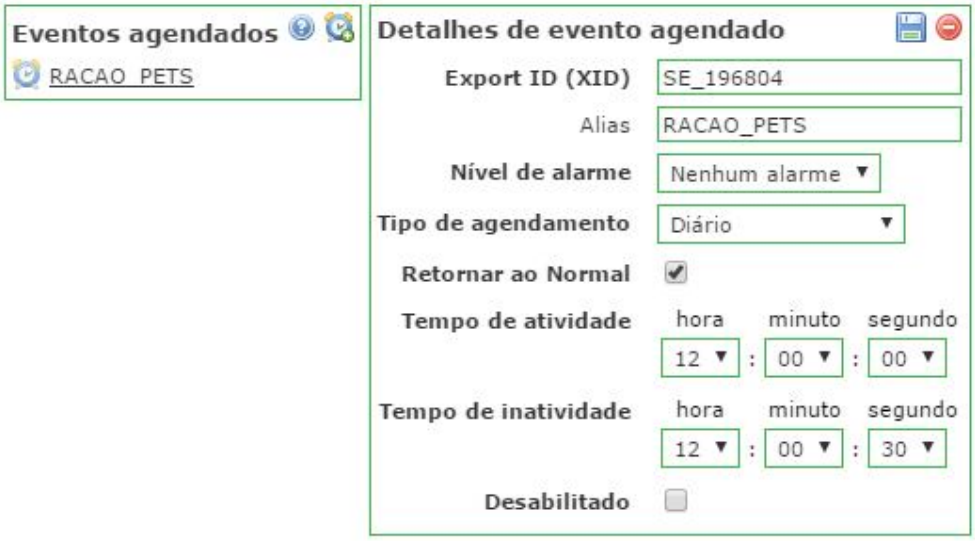


**Figura 15 – Configuração para que evento RACAO_PETS seja acionado diariamente**

Após feita a configuração do evento é necessário criar uma rotina nos tratadores para que o *Data Point* que representará o equipamento seja acionado. Como mostrado na Figura 16, foi definido o tipo como *Set Point* e escolhemos o *Target*, que será o *Data Point* acionado pelo evento que neste caso é o DP_RACAO_PETS onde foi definido o seu valor no caso de atividade como Um (1), que é o estatus de liagado, e na inatividade como Zero (0), que é o estatus de desligado.


**Figura 16 – Configuração para que quando Evento RACAO_PETS seja acionado informe valor para o *Data Point* DP_RACAO_PETS**

## 4.6 BANCO DE DADOS

Referente ao banco de dados foram implementadas duas ações, sendo elas para fechar todas as janelas e desligar as lâmpadas, para isso foram criadas duas *procedures* e uma *trigger*. As tabelas que serão utilizadas para essa implementação são events e pointvalues, como mostrado na Figura 17.

A tabela events recebe informação toda vez que um evento é disparado pela aplicação como, por exemplo, o evento de “CHUVA”, ou seja, cada vez que esse evento é disparado a aplicação insere um registro na tabela events, será utilizado esse registro inserido para saber se é necessário disparar as procedures criadas.

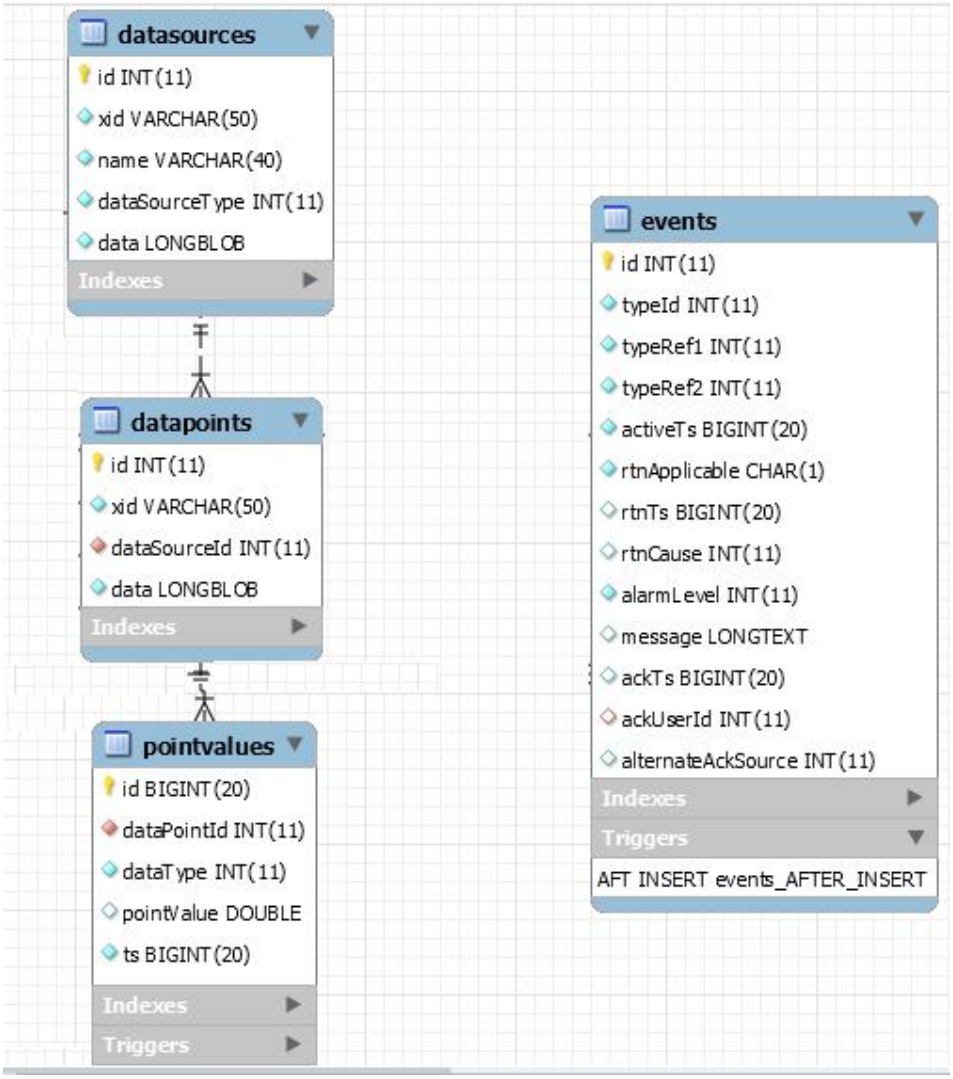


**Figura 17 – Tabelas utilizadas no banco de dados para implementação via banco**

Nos Quadros 1 e 2 serão representadas as implementações das *procedures* criadas, primeiramente para fechar as janelas e posteriormente para desligas às lâmpadas. As duas *procedures* são extremamente parecidas, ambas possuem um parâmetro para definir a ação e então é feito um *Select* buscando todas as Janelas ou Lâmpadas da tabela datapoint e percorrendo todos os registros retornados pelo *Select* e também feito um *Insert* com os campos necessários na tabela pointvalues.

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `FecharAbrir_Janelas`(Abrir  boolean)
BEGIN
declare tipo int(11);
declare idPoint int(11);
DECLARE done INT DEFAULT FALSE;
DECLARE cursor_i CURSOR FOR Select d.id  from scadabr.datapoints d where upper(d.xid) like('%DP_JANELA%');
DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = TRUE;
OPEN cursor_i;
  read_loop: LOOP
    FETCH cursor_i INTO idPoint;
     IF done THEN
      LEAVE read_loop;
     END IF;
if (Abrir = True) then
   set tipo := 1;
end if;

if (Abrir = False) then
   set tipo := 0;
end if;

    INSERT INTO `scadabr`.`pointvalues`(`id`, `dataPointId`, `dataType`, `pointValue`, `ts`)
    VALUES(NULL, idPoint, 1, tipo, (select conv(
        concat(
            substring(uid,16,3),
            substring(uid,10,4),
            substring(uid,1,8))
            ,16,10)
        div 10000
        - (141427 * 24 * 60 * 60 * 1000) as current_mills
from (select uuid() uid) as alias));
  END LOOP;
  CLOSE cursor_i;
END
```

**Quadro 1 - *Procedure* para fechar janela**

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `DesligarLigar_Lampadas`(Ligar  boolean)
BEGIN
declare tipo int(11);
declare idPoint int(11);
DECLARE done INT DEFAULT FALSE;
DECLARE cursor_i CURSOR FOR Select d.id  from scadabr.datapoints d where upper(d.xid) like('%LAMPADA%');
DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = TRUE;
OPEN cursor_i;
  read_loop: LOOP
    FETCH cursor_i INTO idPoint;

     IF done THEN
      LEAVE read_loop;
     END IF;

if (Ligar = True) then
   set tipo := 1;
end if;

if (Ligar = False) then
   set tipo := 0;
end if;

     INSERT INTO `scadabr`.`pointvalues`(`id`, `dataPointId`, `dataType`, `pointValue`, `ts`)
     VALUES(NULL, idPoint, 1, tipo, (select conv(
          concat(
               substring(uid,16,3),
               substring(uid,10,4),
               substring(uid,1,8))
               ,16,10)
          div 10000
          - (141427 * 24 * 60 * 60 * 1000) as current_mills
from (select uuid() uid) as alias));

  END LOOP;
  CLOSE cursor_i;
END
```

**Quadro 2 - *Procedure* para desligar lâmpadas**

Para que as *procedures* que farão a ação de fechar as janelas e desligar as lâmpadas sejam disparadas foi implementada uma *trigger* do tipo *After Insert* na tabela events onde será monitorada a mensagem do evento disparado, ou seja, se na mensagem estiver “CHUVA” será disparado a *procedure* FecharAbrir_Janelas para que as janelas sejam fechadas, e caso a

mensagem do evento seja “CASA_OFF” as duas *procedures* serão disparadas para que sejam fechadas as janelas e desligadas as lâmpadas.

Tais mensagens são as que foram definidas ao criar os eventos ligados aos *Data Points* explicado na seção Eventos associados à Data Points, Eventos Agendados e Tratadores de Eventos.

Abaixo está representada a implementação da *trigger* na tabela events como mostrado no Quadro 3.

```
CREATE DEFINER=`root`@`localhost` TRIGGER `scadabr`.`events_AFTER_INSERT` AFTER INSERT ON `events`
FOR EACH ROW
BEGIN
  if POSITION("CHUVA" in new.message) then
    call scadabr.FecharAbrir_Janelas(False);
  end if;

  if POSITION("CASA_OFF" in new.message) then
     call scadabr.FecharAbrir_Janelas(False);
     call scadabr.DesligarLigar_Lampadas(False);
  end if;
END
```

**Quadro 3 - *Trigger* para acionar *procedures* caso ocorra eventos: “CHUVA” ou “CASA_OFF”**

### Interface de Controle e Monitoramento

Para criar uma interface de controle é possível carregar uma imagem simulando a residência, assim incluindo os componentes/*Data Points* sobre a imagem a interface de controle fica mais fácil de usar e mais agradável visualmente.

Para montar a interface do protótipo desenvolvido foram utilizados os seguintes componentes:

- GIF Binário: Para utilizar um GIF Binário é necessário informar qual *Data Point* estará representando. Existem outras configurações onde usuário pode utiliza-las se desejar como mostrado na Figura 18.

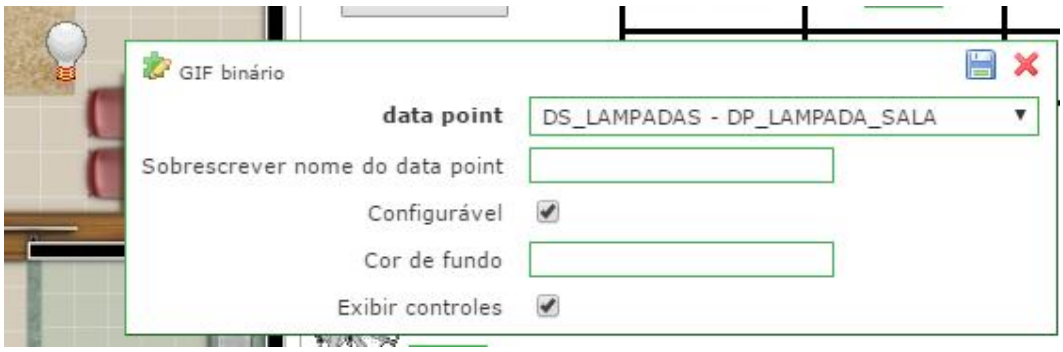


**Figura 18 - Configurações de GIF Binário**

Após configurado o componente é possível configurar imagens referente ao valor do *Data Point*, como se trata de um tipo binário as imagens podem ser configuradas para os valores zero(0) e um(1), como mostrado na Figura 19.

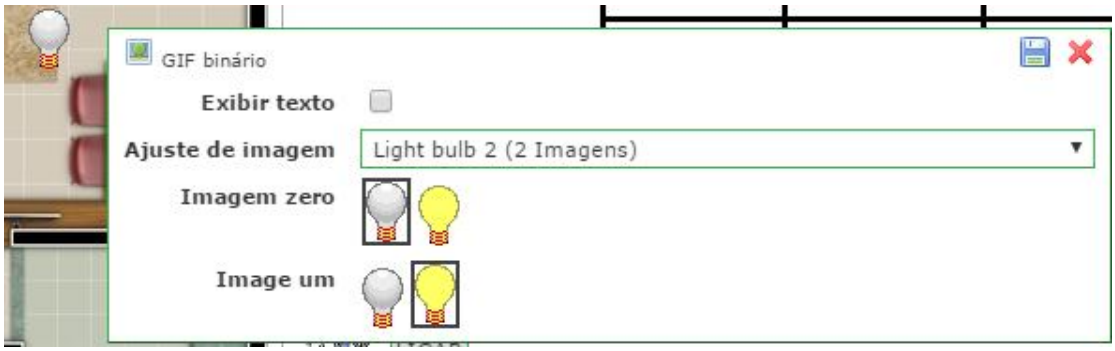


**Figura 19 - Configuração de Ilustração para o GIF Binário**

- *Data Point* Simples: Um *Data Point* Simples não tem a opção de configurar imagens como o GIF Binário, apenas suas configuração são iguais como mostrado na Figura 20.

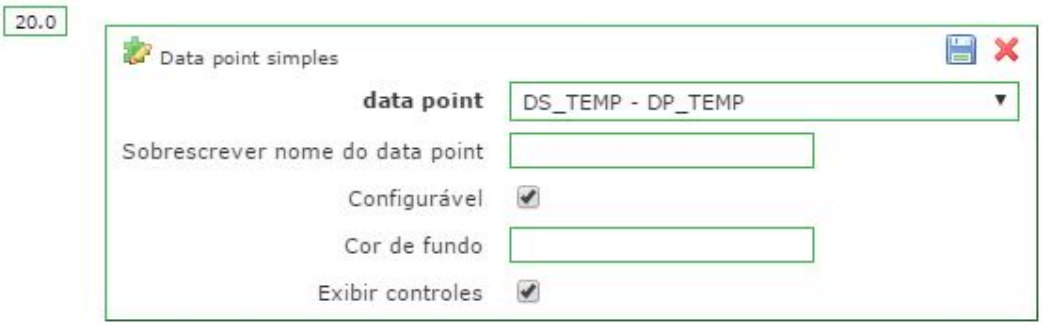


**Figura 20 - Configuração Data Point Simples**

- Botão (Escrita): Botões de escrita são utilizados para redefinir valores aos *Data Points* do tipo Binário como, por exemplo, ligar e desligar uma lâmpada. As configurações consistem em configurar um *Data Point* e definir o texto que apresentará com seu valor igual a zero(0) e um(1), como mostram as Figuras 21 e 22.


**Figura 21 - Configurações Botão Escrita**


**Figura 22 - Configuração de texto Botão Escrita**

- Botão *Script*: As configurações para os Botões *Scripts* são extremamente simples, nelas só é possível definir seu texto e vincular a um *Script* já criado conforma a seção *Scripintg*, como mostrado na Figura 23.


**Figura 23 - Configurações Botão Script**

Na Figura 24 está ilustrado o que cada elemento da Interface corresponde.

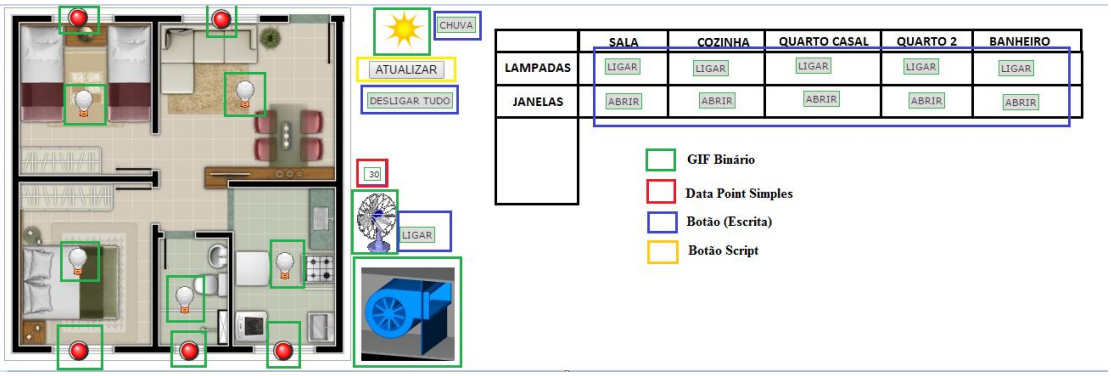


**Figura 24 - Ilustração da Interface**

Assim, através da interface proposta, o usuário pode tanto acompanhar remotamente as condições da moradia quanto interferir no acionamento de equipamentos, se assim for necessário. Essa abordagem, além de atribuir comodidade aos moradores, permite que qualquer sinistro na residência seja informado em tempo real para os responsáveis, talvez evitando infortúnios futuros.

## 5 CONCLUSÃO

Com o desenvolvimento deste trabalho foi possível obter dois tipos de contribuição:

Ele estabelece uma espécie de metodologia de implementação. Muitas vezes, a carência de documentação, característica essa presente no *software* ScadaBR, por exemplo, dificulta a implementação de ferramentas mais sofisticadas no ScadaBR. O presente trabalho contribui nesse sentido ao apresentar, de maneira bastante ilustrativa, os passos necessários para configurar adequadamente inúmeras ações e componentes do ScadaBR focado na automação residencial.

Ele também explora a forma como os dados coletados pelo ScadaBR podem ser manipulados via operações de Banco de Dados. Observa-se que o perfil de usuários que implementa sistemas SCADA é, em geral, voltado para as áreas de engenharia, controle e automação, assim dificultando a exploração de técnicas, ferramentas, tecnologias e conceitos nativos da Ciência da Computação. Então, esse trabalho também colabora nesse sentido, guiando possíveis implementações futuras a usufruir de recursos de Banco de Dados.

Tendo como base os resultados obtidos durante o desenvolvimento deste trabalho e todas as explicações sobre a metodologia de implementação no ScadaBR, seria possível aplica-lo em um protótipo de automação real. Além disso, poderia ser interessante utilizar esse estudo para adaptar a implementação do Banco de Dados para outras áreas, e não apenas focado na automação residencial, como por exemplo na automação industrial, saneamento e entre outras.

**REFERÊNCIAS**

Portal do ScadaBR. **ScadaBR Automação para todos.** <http://www.scadabr.com.br/>.

JUCÁ, S. **A relevância dos softwares educativos na educação profissional**. Revista Ciências e Cognição, Rio de Janeiro, v.8, n.1, p. 22-28, 2006.

SANCHO, J.M. **Para uma tecnologia educacional**. 1. ed. Porto Alegre: Editora ArtMed, 1998.

SILVA, M. R.; OLIVEIRA, R. A.; CARMO, M. J.; JUNIOR, L. O. A.;. **Importância da Ferramenta ScadaBR para o ensino em Engenharia.** In: Congresso Brasileiro de Educação em Engenharia (COBENGE). Gramado – RS, 2013. Disponível em: http://www.ffb.edu.br/sites/default/files/tcc-20082-patrick-romero-frota-quindere.pdf

FARIA, M. H. M.; SILVA, M. R.; OLIVEIRA, A. R.; CARMO, M. J.; JUNIOR, L. O. A.;. **Estudo Comparativo Entre Ferramentas de Supervisão, Controle e Aquisição de Dados e a Importância destas para o ensino em Engenharia.** Congresso Brasileiro de Educação em Engenharia (COBENGE). Belém – PA. 2012. Disponível em: http://www.abenge.org.br/CobengeAnteriores/2012/artigos/104364.pdf

QUINDERÉ, Patrick R. F. **Casa Inteligente – Um Protótipo de Sistema de Automação Residencial de Baixo Custo.** 2009. TCC(Curso de Ciência da Computação), Faculdade Farias Brito, Fortaleza.

MYSQL. **MySQL.** <http://www.mysql.com>.