



**UNIVERSIDADE FEDERAL RURAL DO SEMI-ÁRIDO
PRÓ-REITORIA DE GRADUAÇÃO
CENTRO MULTIDISCIPLINAR DE PAU DOS FERROS
CURSO DE BACHARELADO EM ENGENHARIA DE COMPUTAÇÃO**

**SISTEMA SCADA PARA GERENCIAMENTO DE VAGAS EM
ESTACIONAMENTOS**

JOÃO VITOR GOUVEIA RICARTE

**PAU DOS FERROS - RN
2019**

JOÃO VITOR GOUVEIA RICARTE

**SISTEMA SCADA PARA GERENCIAMENTO DE VAGAS EM
ESTACIONAMENTOS**

Trabalho de Conclusão de Curso apresentado ao Curso de Engenharia de Computação da Universidade Federal Rural do Semi-Árido como pré-requisito para obtenção do Título de Bacharel em Engenharia de Computação.

Orientador: Prof. Dr. Cecílio Martins de Sousa Neto.

**PAU DOS FERROS - RN
2019**

© Todos os direitos estão reservados a Universidade Federal Rural do Semi-Árido. O conteúdo desta obra é de inteira responsabilidade do (a) autor (a), sendo o mesmo, passível de sanções administrativas ou penais, caso sejam infringidas as leis que regulamentam a Propriedade Intelectual, respectivamente, Patentes: Lei nº 9.279/1996 e Direitos Autorais: Lei nº 9.610/1998. O conteúdo desta obra tomar-se-á de domínio público após a data de defesa e homologação da sua respectiva ata. A mesma poderá servir de base literária para novas pesquisas, desde que a obra e seu (a) respectivo (a) autor (a) sejam devidamente citados e mencionados os seus créditos bibliográficos.

R487s Ricarte, João Vitor Gouveia.
SISTEMA SCADA PARA GERENCIAMENTO DE VAGAS EM
ESTACIONAMENTOS / João Vitor Gouveia Ricarte. -
2019.
52 f. : il.

Orientador: Cecílio Martins de Sousa Neto.
Monografia (graduação) - Universidade Federal
Rural do Semi-árido, Curso de Engenharia de
Computação, 2019.

1. SCADA. 2. Automação. 3. Estacionamento. 4.
Mobilidade Urbana. I. Neto, Cecílio Martins de
Sousa, orient. II. Título.

O serviço de Geração Automática de Ficha Catalográfica para Trabalhos de Conclusão de Curso (TCC's) foi desenvolvido pelo Instituto de Ciências Matemáticas e de Computação da Universidade de São Paulo (USP) e gentilmente cedido para o Sistema de Bibliotecas da Universidade Federal Rural do Semi-Árido (SISBI-UFERSA), sendo customizado pela Superintendência de Tecnologia da Informação e Comunicação (SUTIC) sob orientação dos bibliotecários da instituição para ser adaptado às necessidades dos alunos dos Cursos de Graduação e Programas de Pós-Graduação da Universidade.

JOÃO VITOR GOUVEIA RICARTE

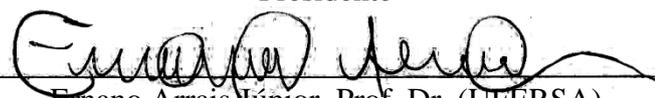
**SISTEMA SCADA PARA GERENCIAMENTO DE VAGAS EM
ESTACIONAMENTOS**

Trabalho de Conclusão de Curso apresentado
ao Curso de Engenharia de Computação da
Universidade Federal Rural do Semi-Árido
como pré-requisito para obtenção do Título de
Bacharel em Engenharia de Computação.

Defendida em: 12/08/2019.

BANCA EXAMINADORA


Cecílio Martins de Sousa Neto, Prof. Dr. (UFERSA)
Presidente


Ernano Arrais Júnior, Prof. Dr. (UFERSA)
Membro Examinador


Rodrigo Soares Semente, Prof. Dr. (UFERSA)
Membro Examinador

Maria de Oliveira Ricarte (In Memoriam)
Pedro Ricarte de Sousa (In Memoriam)

Maria do Socorro Gouveia Brito
Gonçalo Gouveia de Sales
Maria Sueny de Sousa Oliveira
João Bosco Gouveia

AGRADECIMENTOS

Agradeço primeiramente ao Deus todo poderoso que, pela sua misericórdia, contribuiu diretamente com meu discernimento e dedicação a este trabalho de conclusão e ao longo de todo o período deste curso. Pela força e sabedoria concedida que me ajudou a vencer mais uma etapa tão importante na minha vida.

Agradeço aos meus pais João Bosco Gouveia e Maria Sueny de Sousa Oliveira que no empenho, dedicação e força concederam o exemplo para sempre acreditar na minha capacidade e ciência, pelas oportunidades e pela sua ajuda, possibilitaram a conclusão deste curso. Pelo carinho e conforto que sempre esteve presente nas horas que mais necessitei e a que recorria. Aos meus irmãos Vitória e Pedro Ygor que sempre serviu de motivação para buscar algo importante para toda minha família.

Agradeço a toda a minha família Gouveia e a família Ricarte, que acreditaram na importância dessa etapa e souberam reconhecer pelo carinho e palavras de motivação que sempre proferiram no momento de saudades que vim a passar. Em especial a minha tia-madrinha Maria de Fátima Gouveia Brito e minha vó Maria do Socorro Brito Gouveia, que sempre assumiram o papel junto aos meus pais, em me educar e me darem motivação para realizar os meus sonhos. Também agradeço ao meu companheiro Abrantes Junior que me ajudou no meu dia-a-dia até esse momento.

Agradeço ao meu Orientador Cecílio Martins de Sousa Neto, que soube muito bem auxiliar-me e compreender desde o início o propósito do meu trabalho.

“Por vezes sentimos que aquilo que fazemos
não é senão uma gota de água no mar. Mas o
mar seria menor se lhe faltasse uma gota”.
(Madre Teresa de Calcutá)

RESUMO

A superlotação das rodovias por veículos nas cidades torna o trânsito um incômodo aos seus usuários devido à espera para chegar a algum lugar. Na expectativa de fugir dessa espera, os usuários buscam estacionamentos livres e que seja rápido para encontrar uma vaga, mas devido à alta procura por esses locais, é cada vez mais difícil fugir desse incômodo. Esse trabalho apresenta uma solução de gerenciamento de estacionamentos utilizando as técnicas SCADA (*Supervisory Control and Data Acquisition*), propondo uma ferramenta com interface simples e bastante usual para a administração, e um modelo de implementação utilizando o microcontrolador Arduino com uma prototipagem *in loco*, avaliando o projeto e verificando sua usabilidade no ambiente inserido.

Palavras-chave: SCADA. Automação. Estacionamento. Mobilidade Urbana.

ABSTRACT

Overcrowding of highways by vehicles in cities makes traffic a nuisance for its users due to waiting to get somewhere. Expecting to get away from this waiting users are looking for free parking and that is quick to find a parking space, but due to the high demand for these places it is becoming increasingly difficult to escape this annoyance. This work presents a parking management solution using the Supervisory Control and Data Acquisition (SCADA) techniques, proposing a tool with a simple and quite usual interface for administration and an implementation model with a prototyping and in loco, evaluating the project and verifying its usability in the inserted environment.

Keywords: SCADA. Automation. Parking. Urban mobility.

LISTA DE FIGURAS

Figura 1 – Frota de Veículos emplacados	13
Figura 2 - Estacionamento UFERSA - Pau dos Ferros-RN.....	16
Figura 3 – Posição do sensor no local do teste.	16
Figura 4 - Maquete de implementação prática.	19
Figura 5 -Projeto Vagas Simples finalizado.	20
Figura 6 - Representação do protótipo em funcionamento, com todas as vagas ocupadas	21
Figura 7 - Mecanismo automático de regulagem do fluxo de vapor.	25
Figura 8- Modelo do Sistema de Gerenciamento	28
Figura 9 - Placa Arduino Mega 2560	29
Figura 10 - Processo de medição da distância pelo HC-SR04	29
Figura 11 – Sensor HC-SR04	29
Figura 12 - Organizer do Elipse E3	31
Figura 13 - Configuração dos Drive de Comunicação Modbus	31
Figura 14 - Configuração do Drive de Comunicação Modbus.....	32
Figura 15 - Tela de Configuração do Drive e Tags do Elipse E3.....	32
Figura 16 - Teste de Comunicação do Drive	33
Figura 17 - Composição de Quadro de Supervisão	34
Figura 18 - Tela de Estacionamento	34
Figura 19 - Tela de Status da Vaga.....	35
Figura 20 - Associações da Tag.....	36
Figura 21 - Script do Campo Hora de Chegada.....	37
Figura 22 - Tela de Supervisão de Estacionamento	38
Figura 23 - Inserção do Banco de Dados.....	38
Figura 24 - Comunicação do Histórico com as Tags Internas.....	39
Figura 25 - Script do Campo Status Vaga.	40
Figura 26 - Configuração do E3 Browser.....	41
Figura 27 - Tela de Relatório de Vagas.....	41
Figura 28 - Local de Testes <i>in loco</i>	42
Figura 29 - Tela Inicial do Supervisório.....	42
Figura 30 - Tela de Supervisão em funcionamento por eventos	43
Figura 31 - Tela de Relatórios com dados do Supervisório	44

LISTA DE TABELAS

Tabela 1 - Resumo dos trabalhos pela utilização dos dispositivos..... 17

LISTAS DE ABREVIATURAS E SIGLAS

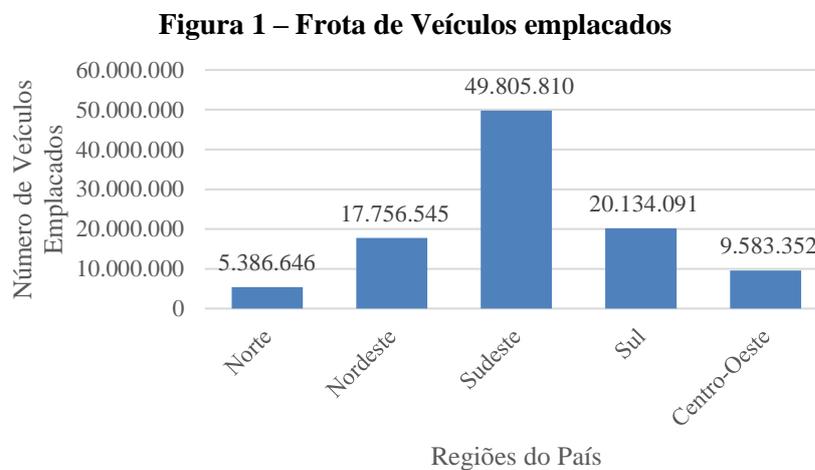
BD	Banco de Dados
CLP	Controlador Lógico Programável
IDE	<i>Integrated Development Environment</i> – Ambiente de Desenvolvimento Integrado
IHM	Interface Homem-Máquina
IoT	Internet of Things – Internet das Coisas
LCD	<i>Liquid Crystal Display</i> – Display de Cristal Líquido
OPC	<i>OLE for Process Control</i> – OLE para Controle de Processo
PWM	<i>Pulse Width Modulation</i> – Modulação de Largura de Pulso
SCADA	<i>Supervisory Control and Data Acquisition</i>
UTR - RTU	Unidade Terminal Remota - <i>Remote Terminal Unit</i>
WEB	<i>World Wide Web</i>
WiFi	<i>Wireless Fidelity</i>

Sumário

Sumário.....	12
1. INTRODUÇÃO	13
1.1 Objetivos.....	15
1.1.1 Objetivos Específicos.....	15
1.2 Metodologia	15
1.3 Organização do Trabalho	16
2. ESTADO DA ARTE.....	17
3. REFERENCIAL TEÓRICO	24
3.1 Mobilidade Urbana	24
3.2 Automação	24
3.3 Sistemas SCADA	25
3.3.1 Sistema Supervisório	25
3.3.2 Software Supervisório.....	27
3.3.2.1 Elipse E3	27
4. RESULTADOS OBTIDOS	28
4.1 Descrição do Sistema.....	28
4.1.1 Microcontrolador	28
4.1.2 Sistema de Sensoriamento	29
4.1.3 Comunicação Elipse E3 – Microcontrolador.....	30
4.2 Protótipo Desenvolvido	33
4.2.1 Telas de Supervisão	33
4.2.2 Relatório.....	38
5. CONSIDERAÇÕES FINAIS.....	45
REFERÊNCIAS.....	46
Anexo A – Código do Sensor Ultrassônico	50
Anexo B – Código do Protocolo Modbus.....	51

1. INTRODUÇÃO

O aumento significativo da frota veicular é uma realidade cada vez mais frequente no cotidiano das cidades brasileiras. Esse crescimento está relacionado com a expansão dos centros urbanos e do poder aquisitivo da sociedade (CASSIANO, 2014). De acordo com o Departamento Nacional de Trânsito, atualmente no Brasil, temos aproximadamente 102,6 milhões de unidades veiculares registradas, sendo divididos por região da seguinte forma: cerca de 17,7 milhões no Nordeste, 5,3 milhões no Norte, 49,8 milhões no Sudeste, 20 milhões no Sul e 9,5 milhões no Centro-Oeste, conforme apresentado na Figura 1 (BRASIL, DENATRAN, 2019).



Fonte: Brasil, Denatran, 2019.

Baseado na Figura 1, essa grande quantidade de veículos nas ruas das cidades brasileiras ocasiona transtornos a circulação diária de indivíduos. Devido a rápida e ineficiente urbanização que não caminhou no mesmo ritmo com os investimentos necessários na infraestrutura, acometem muitos fatores como por exemplo o citado anteriormente (VIANNA, 2013).

O deslocamento na cidade é um dos fatores importantes a influenciar na qualidade de vida do indivíduo. O tempo que se leva de um lugar ao outro, influencia indiretamente, mas significativamente. Isso leva a compreensão que a compreender que o problema da mobilidade urbana no Brasil é de extrema importância, pois nos remete a questões de bem-estar, produtividade e soluções para tais (STEFANELLI e PERO, 2015).

Neste contexto problemático, no cenário rodoviário brasileiro, a solução dos motoristas que evitam passar muito tempo no trânsito é estacionar. A disposição de vagas nos estacionamentos é muito vasta, essa solução é encontrada em quase toda estrutura comercial, de saúde e cultural. Em uma comparação quantitativa, podemos associar que quanto maior o número de vagas, mais veículos podem ocupá-las e quanto mais disponibilidade da

capacidade de vagas, mais suscetível o deslocamento de indivíduos para este lugar (STEFANELLI e PERO, 2015).

Nos grandes estacionamentos pode-se gerar outro agravante em relação ao tempo dentro do veículo. A procura por vagas leva ao motorista percorrer todo o estacionamento, tornando assim mais cansativo e improdutivo seu deslocamento. Disponibilizar vagas a estacionar de maneira rápida e segura é um desafio permanente para os locais públicos e/ou privados. Para se obter um melhor aproveitamento das vagas existentes é necessário um melhor controle e gerenciamento das vagas disponíveis em um determinado estacionamento. O uso da automação de processos favorece essa melhoria (CASSAINO, 2014).

A automação, uma subárea da computação, pode ser definida como uma tecnologia utilizada por meio de sistemas mecânicos, elétricos, eletromecânicos e computacionais na operação e controle da produção, possibilitando a execução do trabalho por meio de máquinas controladas automaticamente (BAYER et. al, 2011). Com o uso de aparelhos que possibilitam o controle e o armazenamento dos dados do processo, associado com as técnicas de controle, propiciam um maior nível de confiabilidade à execução do mesmo (SOUZA, 2005).

Contudo, essa prática em ambientes industriais e comerciais necessitam de um gerenciamento das informações coletadas. Pois à medida que é preciso acessar os dados dos equipamentos, necessita-se de uma aferição direta ao equipamento de controle, o que pode levar a um trabalho cansativo, dependendo da distância e da dificuldade de acesso ao equipamento, sugerindo a necessidade de tornar esses dados disponíveis em um único ambiente (SOUZA, 2005).

Uma solução a essa demanda são os Sistemas de Controle Supervisório e Aquisição de Dados (do inglês, *Supervisory Control and Data Acquisition - SCADA*). Eles oferecem uma solução de supervisão e controle dos processos (BAYER et. al, 2011). O sistema SCADA coleta os dados através do Controlador Lógico Programável (CLP), havendo também a possibilidade de alteração dos dados do processo. Ou seja, os sistemas supervisórios devem dispor e processar os dados e torná-los disponíveis ao operador (SOUZA, 2005). Estas disposições dos dados são realizadas através da IHM (Interface Homem-Máquina). A IHM pode conter botoeiras, interruptores e sinalizações luminosas, bem como painéis digitais, sensível ao toque ou não (FRANÇA, 2015).

Diante disso, a automatização de estacionamentos tem como principal objetivo proporcionar um melhor aproveitamento das vagas existentes e gerenciar o fluxo e a circulação dos veículos. Portanto, o gerenciamento de vagas disponíveis impacta diretamente na qualidade e bem-estar dos condutores de veículos.

1.1 Objetivos

Toda a problemática abordada até agora, revela o quanto procurar uma vaga para estacionar proporciona uma insalubridade ao motorista, por isso, este trabalho, tem como objetivo apresentar uma solução de supervisão para estacionamentos, facilitando assim o gerenciamento e a indicação das vagas disponíveis.

1.1.1 Objetivos Específicos

- Elaborar um sistema de medição compatível para a ferramenta proposta;
- Elaborar a comunicação com o microcontrolador e o sistema de interface;
- Realizar a construção das telas de supervisão para o sistema em questão.

1.2 Metodologia

O presente trabalho foi dividido em etapas para apresentar uma melhor execução, a fim de alcançar o objetivo pretendido. As fases de elaboração do trabalho parte da pesquisa na literatura, seguido da idealização, implementação do modelo, teste da implementação e a aferição dos resultados. A pesquisa na literatura foi direcionada pela similaridade temática do trabalho. Dentro da pesquisa, busca-se a fundamentar a importância do trabalho mostrando a relevância e a popularidade do tema além de adquirir todo embasamento teórico sobre os assuntos apresentados. As pesquisas foram realizadas através de artigos, livros, anais de eventos, trabalhos de conclusão de curso, teses (Mestrado e Doutorado) e fontes na Internet. Em seguida são apresentadas as etapas de idealização do modelo, implementação e teste constituem uma etapa muito importante.

Todo o projeto foi desenvolvido na plataforma Arduino MEGA 2560, por ser uma plataforma de baixo custo e alto nível de implementação para prototipagem, com isso todos os sensores foram escolhidos sensores compatíveis com esta plataforma., juntamente com o ambiente de desenvolvimento do supervisor e da interface, o Elipse E3. Realizando a comunicação entre essas duas plataformas, foi utilizado o protocolo de comunicação MODBUS da empresa Modicon. Esse protocolo é bastante utilizado em comunicações que fazer o uso do Arduino.

Para a aplicação do modelo idealizado foi escolhido um estacionamento na Universidade Federal Rural do Semiárido, Campus Multidisciplinar Pau dos Ferros-RN, localizado em frente ao bloco Administrativo, como exibido na Figura 2.

Figura 2 - Estacionamento UFERSA - Pau dos Ferros-RN.



(a) Visão Frontal - Real

(b) Planta Alta-Projetada

Fonte: UFERSA Pau dos Ferros, Setor de Obras.

Para a medição dos dados foi posto junto ao ambiente um sensor que irá detectar a presença do veículo, como apresentado na Figura 3, e enviar para o sistema de supervisão, assim para o sistema de supervisão e sua interface. A fim de requisitar dados mais relevantes, o supervisor foi posto a execução durante os dias mais movimentados no campus, segunda-feira, 05 de agosto de 2019 no período da manhã, 10:00 às 12:00 horas na instituição.

Figura 3 – Posição do sensor no local do teste.



Fonte: Autor, 2019.

Com os dados coletados foi possível realizar o gerenciamento das vagas e estudar melhorias para o fluxo de veículos, a fim de oferecer condições saudáveis aos motoristas.

1.3 Organização do Trabalho

O seguinte trabalho segue uma organização que facilita a compreensão do tema e dos passos realizados ao longo deste para a construção de um modelo de supervisão para gerenciar estacionamentos e está dividido na seguinte ordem: O capítulo 2 apresenta alguns trabalhos realizados com mesmo eixo temático. Nele sendo descritos os problemas e objetivos de cada trabalho, bem como a solução proposta. O capítulo 3 apresenta o referencial teórico, que contém assuntos relacionados à temática trabalhada nesse documento, com isso esclarecer ao leitor particularidades do tema e de ferramentas utilizadas. No capítulo 4 serão apresentados a descrição do sistema e os mecanismos realizados para sua implementação. Nele também será demonstrado os passos até chegar no protótipo posto à teste. E por fim o capítulo 5 com as considerações finais e sugestões de trabalhos futuros.

2. ESTADO DA ARTE

Nos últimos anos, diversas soluções para o gerenciamento de vagas de estacionamento foram propostas na literatura, aplicados ao gerenciamento de vagas existentes em estacionamentos de ambientes públicos ou privados. Neste capítulo serão apresentadas, 8(oito) destas propostas que serviram como embasamento teórico para a estruturação e execução deste trabalho e na tabela 1 é exposto um resumo sobre os componentes utilizados nestes trabalhos.

Tabela 1 - Resumo dos trabalhos pela utilização dos dispositivos.

<i>Referência</i>	<i>Controlador</i>	<i>Sensor</i>	<i>Interface</i>
REICHERT et al. (2011)	-	-	-
SILVA et al. (2013)	Arduino UNO	Ultrassônico	Sim
LEANDRO et al. (2012)	Arduino UNO	Chaves Reed-Switches	-
FUJITA et al. (2016)	Arduino Mega 2560	LED - LDR	-
BANDEIRA et al. (2014)	Arduino UNO	Ultrassônico	-
CAVAMURA e MITSUHASHI (2014)	Raspberry Pi	Ultrassônico	Sim
VICENTE (2016)	-	Ultrassônico	Sim
RAHMAN e BHOUMIK (2019)	Arduino Mega 2560	Ultrassônico	-

Fonte: Autor, 2019.

O artigo de REICHERT et al. (2011) apresenta uma proposta acerca dos estacionamentos automatizados. O trabalho contextualiza todo o desconforto em passar vários minutos procurando por uma vaga no estacionamento, que faz crescer a demanda por opção de estacionamentos capazes de resolver o problema de uma forma correta, segura e automática, propondo desenvolver uma alternativa para tal problema.

Dado um tema muito recorrente, este artigo inova no algoritmo do sistema computacional de controle, implementando melhorias significativas no processo de tomada de decisões. Este algoritmo gera um banco de dados armazenando informações de todos os veículos estacionados em tempo real. A interface com o usuário permitiria reconfigurar vários atributos do estacionamento não necessitando a reprogramação. A aplicação do conceito estudado pelo trabalho não possui uma validação concreta, apenas a implementação de uma maquete, essa prática torna questionável a aplicação do modelo, dado que as variáveis externas podem influenciar na execução do projeto e afetar na qualidade do mesmo.

No trabalho de SILVA et al. (2013) apresenta-se uma proposta de estacionamento inteligente. Este esclarece que o grande volume de carros em circulação não está diretamente relacionado com o aumento de mobilidade urbana, nem tanto quanto da disponibilidade de vagas para estacionar, logo afetando o dia-a-dia das pessoas. Como observado, a falta de vagas e do conhecimento delas na hora de estacionar torna motivação, tomando como experimento o estacionamento do Campus Teresina do Instituto Federal de Ciência Educação

e Tecnologia do Piauí. Este propõe a automação de todo o estacionamento do campus, criando uma integração das vagas com uma central de controle localizada na guarita do campus. Para realizar a comunicação entre o sistema de controle localizado na guarita, e os sensores, no estacionamento, utilizou um tipo de comunicação sem fio que obtivesse um acesso maleável evitando alterações na estrutura predial do campus, - trazendo essa inovação em comparação com outros trabalhos na literatura. Sendo o projeto pensado para controlar o fluxo de veículos no estacionamento não só do campus de Teresina, mas amenizar esse problema em outros estacionamentos. Obtendo resultados satisfatórios e que comprovaram a eficiência do sistema, tanto de hardware e software, apresentando baixo custo de instalação e fácil manuseio, possui sistema de comunicação eficiente. Ainda no trabalho é proposto como sugestões para futuros projetos, a instalação de um painel onde o próprio usuário obteve-se a informação das vagas disponíveis. Porém o trabalho não apresenta uma proposta de interface para o usuário, sendo insuficiente para a verificação e administração do espaço, falhando assim com a interação mal definida entre o operador e o sistema.

No artigo de LEANDRO et al. (2012) apresenta-se uma outra solução automatizada que viabiliza a busca por vagas em estacionamentos públicos e privados. O trabalho afirma que na automação de estacionamentos há um melhor aproveitamento das vagas existentes, onde é ordenado o fluxo e a circulação dos veículos, além disso, fornece uma completa disposição dos dados gerenciais para o controle eficaz do uso das vagas e da receita do estacionamento.

O trabalho pode ser implementado em qualquer estacionamento de uma instituição de ensino, onde se pode gerenciar as vagas disponíveis com a aplicação do material de caráter simples e de grande utilidade para os que utilizam desses espaços. Neste trabalho é implementado uma maquete cuja estrutura está apresentada na Figura 4. Utilizando a plataforma de desenvolvimento Arduino-UNO, componentes chaveadores denominados Reed-Switches. Com esta interação entre os componentes, percebe-se a redução na quantidade de vagas disponíveis na maquete sendo visualizada automaticamente no LCD do circuito com programação gerada para análise dos níveis das entradas utilizadas. O display LCD representa uma forma prática e de baixo custo para exibir as informações.

A utilização da plataforma Arduino proporciona uma considerável confiabilidade aos resultados obtidos com uma margem de erro muito pequena em relação com as que utilizam portas lógicas básicas. Sendo de fácil compreensão e viabilidade financeira sobre o projeto proposto. Trazendo essa proposta em maquete o dimensionamento desse projeto em uma estrutura física, como mencionado anteriormente pode acarretar muitos outros

problemas, tanto para estrutura do local, quanto para a do sistema. O projeto também não apresenta nenhuma solução de supervisão, onde a informação pode ser centralizada e mais bem utilizada para a administração do estacionamento.

Figura 4 - Maquete de implementação prática.



Fonte: LEANDRO et al. (2012).

No trabalho de FUJITA et al. (2016), é proposto uma ferramenta que tem como objetivo reduzir o tempo gasto na procura por uma vaga para estacionar e a elaboração de uma interface, onde se encontra indicações da disponibilidade das vagas para o usuário. A justificativa para o uso dessa ferramenta, seria a redução do tempo gasto, já que os usuários já estariam cientes de onde haveria vagas disponíveis antes de entrar no local.

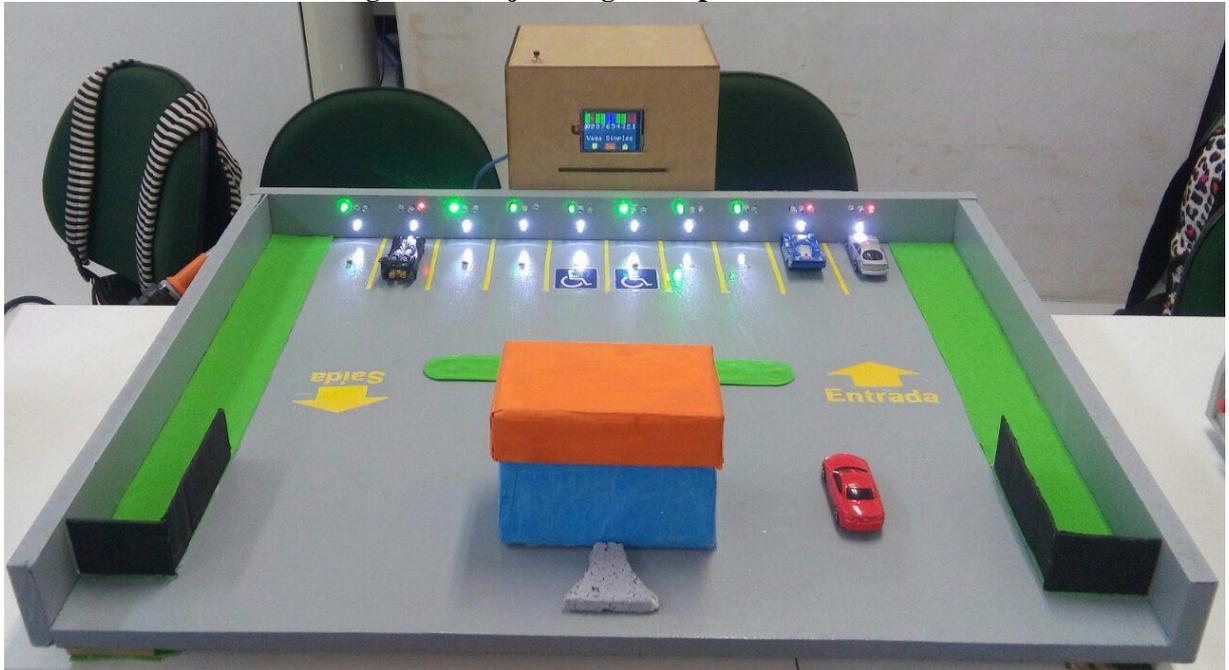
Para solucionar a problemática, salientada pelo artigo, onde os motoristas levariam tempo para encontrar uma vaga para estacionar, de modo que ao chegar no estacionamento já saibam a localização de vagas livres, optando pela automação do estacionamento para atender as necessidades. Tratando como objetivo a elaboração de uma interface gráfica que indica quais as vagas disponíveis para os usuários e onde elas estão localizadas e a escolha das vagas por meio do toque na tela da interface.

Dados a recorrente pesquisa ao assunto tratado, o seguinte trabalho tem como diferencial a possibilidade da escolha da seleção da vaga, antes de adentrar ao estacionamento, evitando disputas por parte dos usuários. O projeto utiliza como base da programação a plataforma em Arduino e dispõem de maquetes para a exemplificação do modelo, ilustrado na Figura 5.

O trabalho exposto conseguiu implementar e obter seus objetivos e verificando que a partir do protótipo a implementação em um ambiente real haveria uma diminuição do tempo gasto na procura de vagas disponíveis. No entanto, melhorias como uma interface que possa

conter mais informações, como hora de chegada e hora de saída, bem como o armazenamento dessas informações em um banco de dados poderiam aperfeiçoá-lo, tornando mais usual a ser aplicado em diferentes estacionamentos.

Figura 5 -Projeto Vagas Simples finalizado.



Fonte: FUJITA et al. (2016).

No trabalho de BANDEIRA et al. (2014) é apresentado um modelo protótipo para o gerenciamento de um estacionamento automatizado. O trabalho afirma que uma realidade do cotidiano na população é o aumento da frota de veículos, crescente a cada dia oferecendo uma dificuldade aos motoristas em encontrar vagas disponíveis nos estacionamentos.

O trabalho oferece uma solução com os conceitos de eletrônica e computação afim de operar e gerenciar através de procedimentos automatizados, utilizando um modelo computacional matricial. Produzindo um protótipo, ilustrado na Figura 6, utilizando a plataforma de programação Arduino para localizar e contabilizar as vagas disponíveis com o auxílio de sensores e atuadores luminosos. Tendo como objetivo desenvolver um método capaz de gerir a circulação de automóveis a procura de vagas dentro do estabelecimento.

O sistema de detecção das vagas foi implementado utilizando a prototipagem Arduino, onde foram feitos testes em ambientes simulados utilizando uma estrutura modelo. Foi possível notar que conforme a quantidade de vagas ocupadas o número de vagas livres é disposto no display localizados a estrada do estacionamento.

O artigo salienta que o sistema foi testado inúmeras vezes de forma segura e confiável, onde o percentual de erros encontrados no sistema como um todo são relativamente pequenos, sendo possível e aplicável. O protótipo desenvolvido atingiu todos os objetivos descritos

inicialmente no artigo, conciliando os conceitos eletrônicos e computacionais a fim de automatizar o estacionamento garantindo o conforto e agilidade no fluxo dos veículos. Apesar do sistema apresentar soluções viáveis para a solução, ela não oferece um suporte com banco de dados e uma interface que auxilie no gerenciamento do estacionamento.

Figura 6 - Representação do protótipo em funcionamento, com todas as vagas ocupadas



Fonte: BANDEIRA et al. (2014).

O trabalho de CAVAMURA e MITSUHASHI (2014) apresenta uma solução para o gerenciamento de vagas disponíveis para estacionamentos de veículos. O trabalho afirma que as soluções existentes no que se diz respeito ao gerenciamento de vagas não atendem a totalidades das funcionalidades requisitadas, necessitando sempre de algo a mais. Demonstrando que os gerenciadores existentes apenas demonstram a indicação da presença do carro sobre as vagas, não existindo um monitor antes de entrar no estacionamento, verificando qual a vaga e onde ela está situada, obrigando ao usuário percorrer todo o estacionamento a procura de estacionar. Propondo em gerenciar o estacionamento ao usar sensores instalados em cada uma das vagas visando verificar o estado da vaga que auxiliará no ato de estacionar, fixo na parede e possuindo um alarme sonoro e luminoso sendo acionado e acentuando conforme o veículo se aproxima do limite estabelecido.

O projeto também utilizou os conhecimentos de hardware e software, imergindo nos conhecimentos de comunicação sem fio, além dos estudos dos materiais que apoiariam ao desenvolvimento como o sensor de distância, microcontrolador e a Raspberry Pi. Alcançando

o objetivo técnico de desenvolver o produto, um conjunto de sensores centrais e ambos comunicando entre si, utilizando a comunicação sem fio e baterias, que segundo o autor facilitou na sua implementação, pois a utilização dos cabos poderia alterar a infraestrutura do estacionamento.

No trabalho foi ressaltado algumas dificuldades no desenvolvimento, como a utilização do Raspberry Pi, na compatibilidade das linguagens de programação e na capacidade da placa; também na compreensão dos conceitos de transmissão de dados entre os módulos. Apesar das dificuldades existentes o sistema apresentou-se muito eficiente para sistemas de pequeno porte, conseguindo mostrar em tempo real os estados das vagas do estacionamento. A rede sem fio consegue realizar as configurações nos módulos, enviar de forma correta os estados das vagas até a central e reestabelece a conexão da rede sem haver a necessidade de reconfiguração. Utilizando-se da interface gráfica, esta mostrou fácil operação ao motorista possibilitando a visualização das vagas em uma planta do estacionamento real. Apesar das inúmeras ferramentas para o trabalho, a proposta não conta com um sistema que facilite a administração do estacionamento e nem o armazenamento de informações que podem ser relevantes para o gerenciamento do local.

Em VICENTE (2016) é trazido uma implementação dos conceitos de *IoT* (Internet of Things – Internet das Coisas), um conceito em que as conexões entre diversos dispositivos conectados à rede para coletar, analisar e distribuir dados transformando-as em informações relevantes a serem utilizadas pelo usuário ou manipuladas pelo sistema, contextualizando a temática à um cenário cotidiano, como a mobilidade urbana. Como aponta o trabalho, onde os brasileiros sempre optaram pelo modelo de transporte rodoviário, principalmente àqueles que vivem nas grandes metrópoles

Sabemos das dificuldades de se conseguir uma vaga para estacionar, contudo, o trabalho traz como solução uma alternativa com o propósito de controle inteligente de vagas para estacionamento. Simulando uma arquitetura completa de *IoT*, desde a camada de sensoriamento até a camada de aplicação, assim dividindo o trabalho em partes. A primeira tem como objetivo a integração dos sensores ao microcontrolador, conectado à Internet via WiFi, conectado ao broker MQTT onde serão publicados os pacotes de dados gerados pelo sensor. A segunda fase do projeto é a criação de uma página na Web onde o usuário poderá reservar e acompanhar as vagas do estacionamento.

Com a criação deste protótipo apresenta-se uma alternativa ainda mais automatizada, onde utiliza-se um sensor para a verificação da presença de carro na vaga e possibilita a comprovação da reserva pelo próprio usuário. Analisando os objetivos propostos pelo

trabalho, seu desenvolvimento e resultados pode-se concluir que um protótipo funcional foi criado e que este pode ser escalado e melhorado tanto para trabalhos futuros como para aplicação na forma de um produto comercial. Levantando a problemática da dependência do uso da internet para o bom funcionamento do sistema, podendo ocasionar a perda dos dados caso algum imprevisto aconteça.

No artigo de RAHMAN e BHOUMIK (2019) aponta o rápido aumento do congestionamento no tráfego, devido a escolha de mais veículo particulares do que os veículos públicos, dificultando a busca por espaços para estacionar nas áreas metropolitanas durante as horas mais agitadas. O projeto propõe um sistema de estacionamento inteligente que oferece informações para as pessoas que encontram uma vaga para estacionar online. Facilitando em relação ao tempo de procura e com a escolha da vaga a ser ocupada. Ainda sim em alguns países que já oferecem informações sobre as áreas de estacionamento através da internet, onde é fornecido a informação sobre o local da vaga, mas não é capaz de dizer se ela está ocupada ou não.

Demonstrando assim a necessidade de encontrar rapidamente o caminho para a vaga de estacionar. Neste trabalho, o Arduino é o “cérebro” para o sistema, controlando e vigiando os componentes, com o ultrassônico, colocados nas vagas do estacionamento. Um sensor é adicionado na entrada do estacionamento, onde ao ser acionado o Arduino passa a verificar uma vaga disponível no estacionamento e abrindo a cancela do estacionamento. O sistema será conectado à Internet através do cabo Lan e enviar os status do sistema, podendo ser visualizado facilmente.

Após os testes os autores puderam concluir que a aplicação é totalmente realizável, sendo os principais benefícios: a economia de tempo e combustível, fornecimento de gerência ao estacionamento, reduz os incômodos nos estacionamentos e engarrafamento. Apresentando alguns perigos como o uso da internet, caso a conexão venha a falhar, um arquivo onde são armazenadas todas as informações necessárias para o gerenciamento do estacionamento, bem como uma interface centralizada.

De forma geral todos os trabalhos apresentam um modelo que auxiliam ao motorista o direcionando à vaga de estacionamento que não está ocupada, sendo insuficiente a informação para os setores administrativos do estacionamento que também necessitam de informações sobre o fluxo das vagas ocupadas e/ou livres de forma a melhorar a gerência do estabelecimento, onde uma interface que seja simples e ao mesmo tempo funcional para o operador do sistema, torna-se um requisito fundamental para o projeto.

3. REFERENCIAL TEÓRICO

3.1 Mobilidade Urbana

O crescimento populacional desordenado cria um ambiente arriscado à saúde dos indivíduos, conseqüentemente o expressivo aumento do número dos transportes individuais levam a questão da mobilidade a ser prejudicada pelos acidentes de trânsito, congestionamentos e emissão de poluentes na atmosfera (BRASIL, IPEA, 2016).

Vários fatores influenciam na aplicação correta da mobilidade nos espaços urbanos. Na cartilha do Ministério das Cidades, mobilidade urbana é definido como:

“[...] um atributo das cidades e se refere à facilidade de deslocamentos de pessoas e bens no espaço urbano. Tais deslocamentos são feitos através de veículos, vias e toda a infraestrutura (vias, calçadas, etc.) que possibilitam esse ir e vir cotidiano. Isso significa que a mobilidade urbana é mais do que o que chamamos de transporte urbano, ou seja, mais do que o conjunto de serviços e meios de deslocamento de pessoas e bens. É o resultado da interação entre os deslocamentos de pessoas e bens com a cidade” (BRASIL, MINISTÉRIO DAS CIDADES, 2005, p.3)

Levando ao entendimento que o transporte não é uma única e exclusiva solução para os problemas de mobilidade, mas também uma forma de como a população pode acessar os destinos desejados na cidade. Sendo assim, um conjunto de modos, redes e infraestruturas que garantem a locomoção no espaço urbano mantendo fortes interações com os outros serviços oferecidos (BRASIL, IBAM, 2005).

No entendimento da necessidade de uma política de mobilidade faz necessário ter um conjunto de princípios e diretrizes que possam orientar ações públicas e, posteriores, reivindicações da população. O não cumprimento, ou má execução, dessa necessidade faz com que os indivíduos busquem suas próprias soluções de deslocamento, possibilitando mais riscos e priorizando, o acesso, aos que possuem mais recursos (BRASIL, MINISTÉRIO DAS CIDADES, 2005).

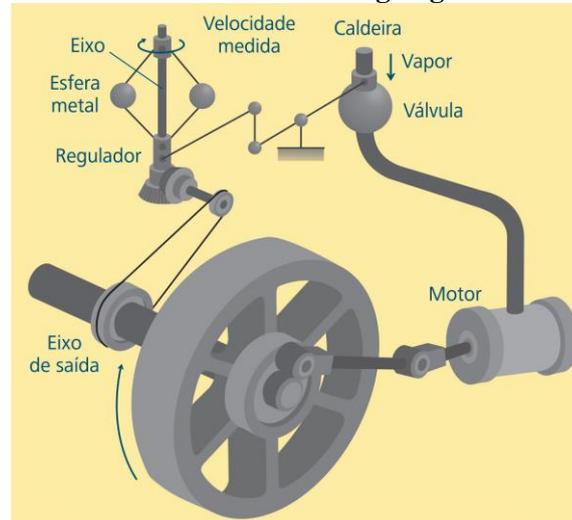
3.2 Automação

Podemos definir a automação como a substituição do trabalho humano e/ou animal por uma execução de máquina, de um sistema automaticamente executado controlado de forma remota, com a mínima interferência humana, ou seja, a automação é “o controle de processos automáticos” (RIBEIRO, 2003). O uso da automação na indústria tem sido causa de aumento na qualidade e quantidade de produtos, oferecendo preços atrativos, aumentando a eficiência dos processos realizados (BARROS, 2012).

A procura por essa eficiência nos processos levou ao começo de uma era de inovações, como citado por BAYER *et al.* (2011, p. 16): “Por volta de 1788, James Watt desenvolveu um mecanismo de regulagem do fluxo de vapor em máquinas”, considerado um dos primeiros

sistemas de controles com realimentação, exibido na Figura 7. No século XX, a automação passou a utilizar os computadores, servomecanismos e controladores programáveis.

Figura 7 - Mecanismo automático de regulação do fluxo de vapor.



Fonte: Roggia e Fuentes (2016).

A automação pode ser aplicada em todas as diversas atividades do cotidiano, por isso ela recebe várias classificações, como, automação bancária, comercial, agrícola, de comunicações e de transporte. Além dessas classificações podemos também encontrar a automação nos processos industriais, ou como se classifica, automação industrial (BAYER et al., 2011). Na automação industrial são criados mecanismos realizáveis à produção com eficiência ao menor custo, como por exemplo, o melhoramento da produtividade das condições de trabalho, a realização das operações, que seriam impossíveis de um indivíduo controlar (SILVEIRA, 2013).

3.3 Sistemas SCADA

No processo da industrialização e dos avanços tecnológicos as práticas manuais tornaram-se ineficientes, por não oferecem os resultados e a padronização que as produções necessitavam, aderindo à prática da automatização desses processos. Contudo, a distância entre os equipamentos e todo o aparato do sistema torna o trabalho de difícil operação e acesso. Os sistemas supervisórios resolvem esta necessidade, permitindo coletar as informações do processo e ainda monitora e atuar sobre este (SOUZA, 2005).

3.3.1 Sistema Supervisório

O SCADA, para PEREIRA e PARDI (2003), deve ser capaz de processar os dados e torná-los disponíveis para o operador do processo ou qualquer outro usuário do sistema de supervisão. Este deve atuar sobre o sistema, dado a ciência do operador. Permitindo o monitoramento e rastreamento de informações do processo, coletados através de

equipamentos de aquisição de dados (KRETZER e AZEVEDO, 2013). Podemos listar os principais objetivos de um sistema supervisório (*Inductive Automation*, 2018):

- Manter o controle de processos industriais localmente ou em locais remotos;
- Monitorar, reunir e processar dados em tempo real;
- Interagir diretamente com dispositivos como sensores, válvulas, bombas, e motores por meio da interação IHM;
- Registrar eventos em um arquivo log.

Os componentes dos sistemas supervisórios podem ser físicos e lógicos. Os componentes físicos podem ser resumidos em sensores e atuadores, redes de comunicação, estações remotas e de monitoramento central. Os sensores são aparelhos conectados aos controladores e monitorados pelo sistema, convertendo parâmetros físicos em sinais analógicos e digitais pela estação remota, já os atuadores são utilizados para atuar sobre o sistema, como o próprio nome ressalta, ligando e desligando coordenados equipamentos. A rede de comunicação do sistema é o meio por onde as informações fluem dos controladores para o Sistema SCADA (KRETZER e AZEVEDO, 2013). Dado o processo de controle e aquisição dos dados, inicia-se nas estações dos CLPs ou Unidades Terminais Remotas (UTR). Esses dois, são microcomputadores que comunica uma série de objetos, sensores e atuadores, e encaminham essas informações para os computadores como o SCADA (*Inductive Automation*, 2018).

Os componentes lógicos dividem suas funções em blocos, permitindo uma maior ou menor flexibilidade e robustez nos parâmetros da aplicação, onde pode-se listar alguns desses blocos: Núcleo de processamento; Comunicação com CLPs/UTRs; Gerenciamento de Alarmes; Históricos e Banco de Dados; Lógicas de Programação Interna (Scripts) ou Controle; Interface Gráfica; Relatórios; Comunicação com outras estações SCADA; Comunicação com Sistemas Externos/Corporativos. A primícias do sistema de supervisão é a comunicação dos equipamentos de campo, de forma descritiva podemos considerar que o Núcleo de Processamento distribui e coordena o tráfego dos dados/informações para os demais módulos do sistema, até chegar à IHM, a fim de mostrar os estados dos dispositivos e consequentemente do processo, informando eventuais problemas dentre outras informações necessárias ao processo (OPC, 2012) conforme citado por KRETZER e AZEVEDO (2013).

Como esclarecida do em KRETZER e AZEVEDO (2013), os sistemas de supervisão e aquisição de dados utilizam um software para monitorar e supervisionar as variáveis e dispositivos dos sistemas de controle conectados aos controladores.

3.3.2 Software Supervisório

Segundo CORREA (2012) o software supervisório demonstra as ocorrências do processo através de telas, configuradas conforme as entradas destinadas ao controlador lógico. Inspeccionando os sinais lógicos vindo dos sensores, recebidos pelo controlador. Com essa comunicação, controlador-supervisório, torna-se viável intervir em qualquer sistema automatizado, sendo a comunicação em alto nível entre o operador e o processo, um dos objetivos do software supervisório.

3.3.2.1 *Eclipse E3*

O sistema Eclipse E3 é uma plataforma SCADA que oferece suporte as aplicações avançadas e distribuídas, ideal para sistemas críticos e centros de controle. Esta ferramenta auxilia no monitoramento e controle de processos, oferecendo uma constante evolução para as mais diversas aplicações, desde interface simples até aos complexos de operação em tempo real. O sistema E3 permite uma conexão com a maioria dos equipamentos, reduz o tempo de desenvolvimento e manutenção das aplicações, integra sistemas corporativos e de gestão. As informações do sistema podem ser manipuladas de diversas maneiras, gerenciar alarmes, armazenamento em Banco de Dados, comunicações entre outros softwares e componentes, além da usabilidade de relatórios (Eclipse, 2019).

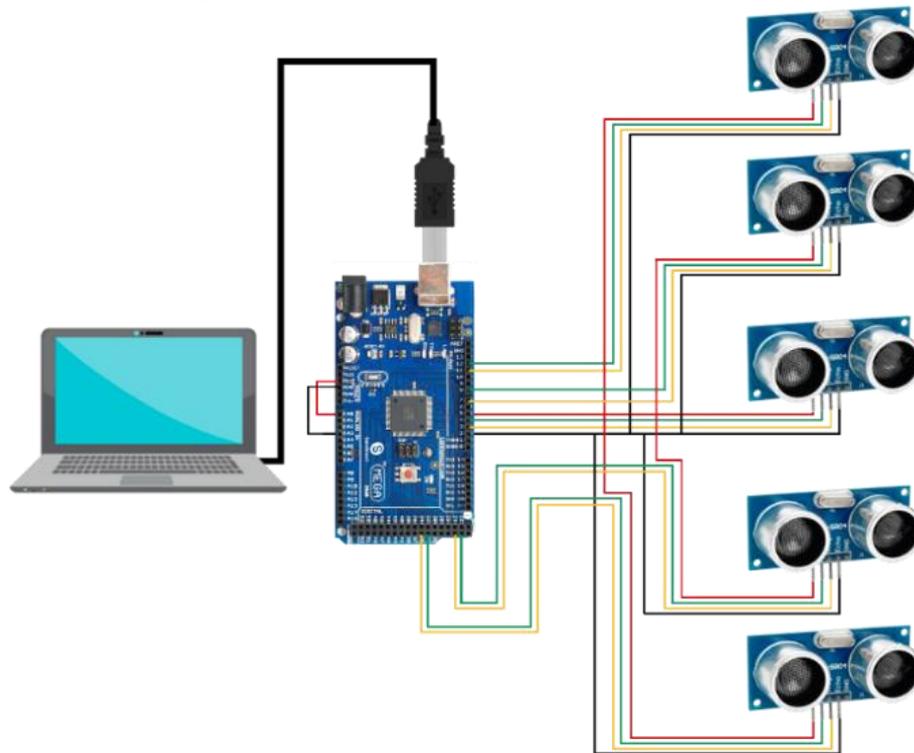
O Eclipse E3 pode ser dividido em três programas, em relação à vista do usuário: E3 Sever, E3 Viewer e o E3 Studio. O E3 Sever é o servidor de aplicações, onde são processadas as comunicações e gerenciados os principais processos do sistema. Comunicando assim, informações gráficas e dados para os clientes em qualquer ponto da rede. O E3 Viewer é a interface de operação com o usuário (console), permitindo executar a aplicação que está no servidor em qualquer ponto da rede. Não sendo necessário a instalação do aplicativo na máquina cliente, pois os componentes são baixados do servidor e registrados automaticamente. E por último o E3 Studio, essa ferramenta de configuração do sistema, serve como plataforma de desenvolvimento. Ela possui um ambiente inclui a configuração da comunicação e um editor gráfico para a criação de telas de operação e scripts integrados, permitindo a reedição por várias pessoas ao mesmo tempo (Eclipse, 2019).

4. RESULTADOS OBTIDOS

4.1 Descrição do Sistema

O modelo do sistema para o gerenciamento de um estacionamento, por meio da supervisão de processos é apresentado na Figura 8. Ele é composto por um microcontrolador Arduino, que fará a aquisição dos dados dos sensores, o sistema de sensoriamento composto por cinco sensores de distância e para fazer a interação entre o microcontrolador e a interface foi utilizada comunicação serial MODBUS. A partir desses componentes podemos realizar o monitoramento do estacionamento. Todo o processo de descrição do sistema é discorrido durante todo o capítulo.

Figura 8- Modelo do Sistema de Gerenciamento



Fonte: Autor, 2019.

4.1.1 Microcontrolador

O microcontrolador utilizado para a implementação do projeto foi o Arduino Mega 2560. O Arduino é uma plataforma eletrônica *open source* de fácil programação e montagem de seus dispositivos. As placas Arduino são capazes de ler entradas e transformá-las em saídas, isso através de instruções passadas ao microcontrolados utilizando a linguagem de programação Arduino e a sua plataforma de edição de código o *Software* Arduino (IDE).

A placa Arduino utilizado no projeto é o Arduino Mega 2560, apresentado na Figura 9. Essa placa possui ao todo 54 pinos de entrada/saída digitais, das quais 15 podem ser usadas

como PWM (*Phase Width Modulation* – Modulação de Largura de Pulso), 16 portas analógicas, 4 portas seriais de hardware e um oscilador de cristal de 16 MHz (Arduino, 2019).

Figura 9 - Placa Arduino Mega 2560

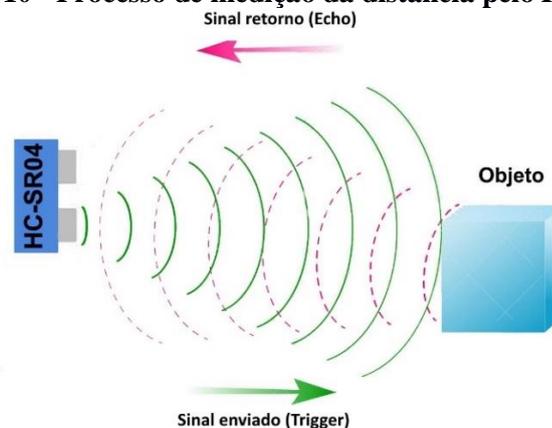


Fonte: Autor, 2019.

4.1.2 Sistema de Sensoriamento

O sistema de sensoriamento é formado apenas pelo sensor de distância HC-SR04. Este sensor é um componente que permite a leituras de distâncias de 2 cm a 4 m, com precisão de 3 mm. O funcionamento do sensor baseia-se no envio de sinais ultrassônicos pelo sensor, que aguarda o retorno do sinal e pelo cálculo do tempo de envio e retorno, a distância do sensor para o objeto é determinada, esse processo é ilustrado na Figura 10 (THOMSEN, 2011).

Figura 10 - Processo de medição da distância pelo HC-SR04



Fonte: THOMSEN, 2011.

O sensor, demonstrado na Figura 11, possui 4 (quatro) pinos, sendo: uma alimentação de 5 V, o pino TRIG que emite o sinal ultrassônico, o pino ECHO que recebe o sinal e o pino GND (THOMSEN, 2011).

Figura 11 – Sensor HC-SR04



Fonte: Autor, 2019.

Primeiramente um sinal é enviado um pulso de 10 μ s, iniciando a transmissão de dados, depois, são enviados 8 pulsos de 40 KHz esperados pelo retorno, determinando assim a distância entre o sensor e os objetos, utilizando a equação $Distância = (Tempo\ ECHO\ em\ nível\ alto * velocidade\ do\ som) / 2$. Para a utilização do Sensor utiliza-se a biblioteca *Ultrasonic*. Descompactando-a e inserindo na pasta “*Libraries*” do Arduino, terá condições de utilizar o sensor (THOMSEN, 2011). O Código de utilização está no Anexo A.

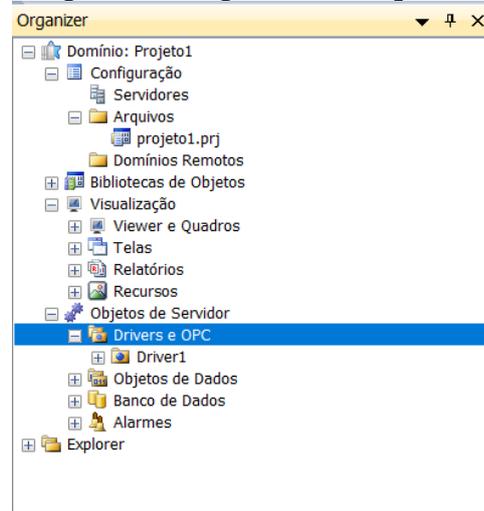
4.1.3 Comunicação Elipse E3 – Microcontrolador

A comunicação entre o microcontrolador e o sistema de supervisão é realizada através de um protocolo de comunicação, onde as informações coletadas nos sensores do Arduino é levado para as o sistema Elipse E3. O protocolo escolhido para a comunicação do presente é o MODBUS. Esse protocolo foi criado nos anos de 1979 pela Modicon. Sendo um composto de mensagens utilizadas para configurar uma comunicação entre dispositivos mestre-escravo, ou cliente/servidor. A comunicação desse protocolo ocorre de forma que o dispositivo mestre inicie as transmissões e os dispositivos escravos respondam de acordo com a requisição do mestre, onde os escravos processam as informações e submetem ao dispositivo mestre (JUNIOR e GRUNEWALDER, 2013).

Para a utilizar o protocolo deve ser configurado nas duas pontas da comunicação, no microcontrolador e no sistema de supervisão. Primeiramente, a configuração no microcontrolador é realizado apenas nas linhas de programação. A biblioteca utilizada para a comunicação foi a “*SimplesModbusSlaveV10*” e inserindo na pasta “*Libraries*” do Arduino. Com isso referenciamos nas linhas de código a utilização do arquivo “*SimpleModbusSlave.h*”, por seguinte a criação das variáveis que serão responsáveis pelas transferências dos dados vindos dos sensores no campo “*enum*”, onde é armazenado os valores das variáveis do sistema externo. Na função “*setup*” do Arduino é configurado o protocolo de comunicação com os comandos “*modbus_configure(&Serial, 9600, SERIAL_8N1, 1, 2, HOLDING_REGS_SIZE, holdingRegs)*” que representa a parametrização da comunicação. Os parâmetros: segundo, terceiro e quarto que dizem respeito à taxa de transmissão de dados, o formato do pacote utilizado no protocolo *Modbus* e a identificação do escravo, respectivamente; e a função “*modbus_update_comms(9600, SERIAL_8N1, 1)*” que também é responsável pela comunicação via *Modbus*. Todo o processo da utilização do protocolo é demonstrado no Anexo B. Este processo acontece de um lado da comunicação, no entanto para haver a real comunicação é necessário que os dois lados estejam configurados com o mesmo protocolo.

Para a realização dessa configuração é necessário o arquivo do protocolo “Modbus.dll” baixado do site da Elipse. Após o início da aplicação deve-se abrir a seção no *Organizer* do Elipse E3, exposto na Figura 12, e buscar por “Drivers e OPC” e com o clique direito, inserir um Driver de Comunicação e selecionar o projeto.

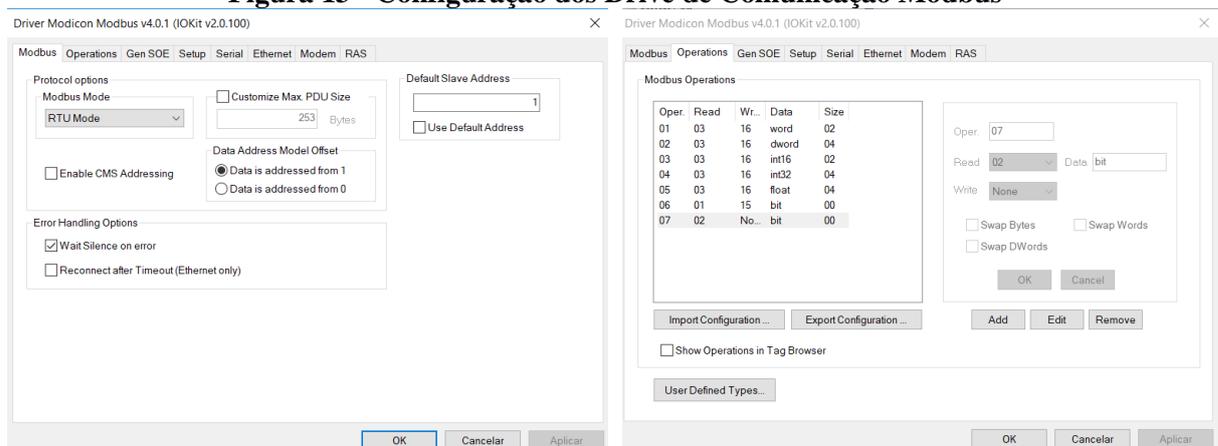
Figura 12 - Organizer do Elipse E3



Fonte: Autor, 2019.

Após a inserção da seção de Driver, podemos configurar o Drive de comunicação. Ao procurar pelas configurações do drive devemos selecionar no campo “Modbus Mode” para “RTU Mode”, exposto na Figura 13(a). Na aba “Operations” deve configurar as operações de acordo como ilustrado pela Figura 13(b). As operações apresentam algumas predefinições, onde a operação 01 é utilizada para as saídas digitais, a operação 02 é utilizada para as entradas digitais, a operação 03 para entradas analógica e operação 04 para saídas analógicas.

Figura 13 - Configuração dos Drive de Comunicação Modbus



(a) Aba Modbus

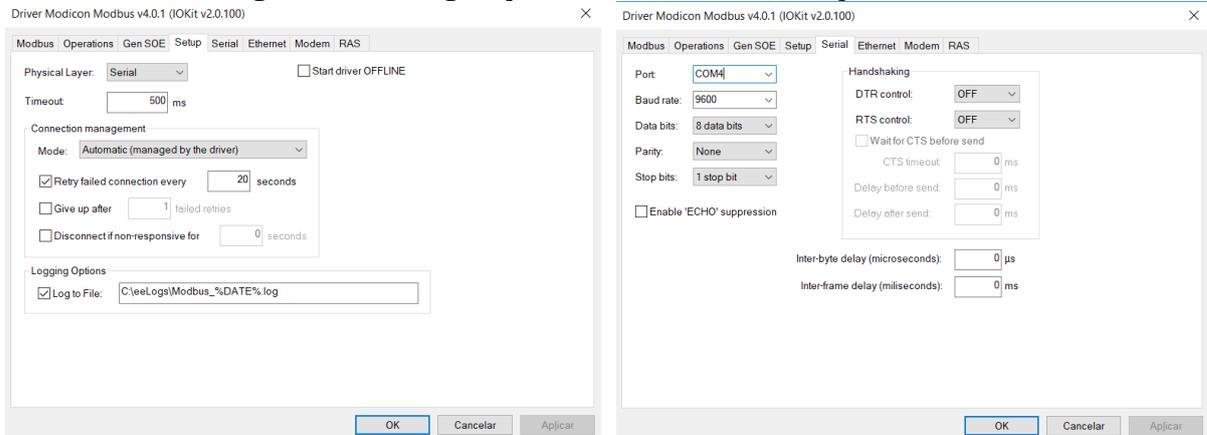
(b) Aba Operations

Fonte: Autor, 2019.

Na aba “Setup” em “Physical Layer” selecione a opção “Serial” com “Timeout” de 500 ms, ilustrado na Figura 14(a). Na aba “Serial” em “Port” selecione a porta na qual o

dispositivo está conectado, nesse caso a porta de conexão do Arduino, e configurando com as informações apresentadas na Figura 14(b).

Figura 14 - Configuração do Drive de Comunicação Modbus



(a) Aba Setup

(b) Aba Serial

Fonte: Autor, 2019.

Ao configurar o driver, para iniciar a conexão deve ser criada tags que receberam a comunicação das variáveis criadas no microcontrolador, exposto na Figura 15. As configurações das Tags, são realizadas através dos campos N1, N2, N3 e N4, que sequencialmente representam: O endereço do equipamento escravo (Arduino), o código da operação, o parâmetro adicional e o endereço do registro no Arduino, na ordem que foi escrita no código.

Figura 15 - Tela de Configuração do Drive e Tags do Elipse E3

Nome	Disp...	Item	P1/N...	P2/N...	P3/N...	P4/N...	Ta...	Va...	Leitur...	Escrita...	Escala?	Min...	Máx...	UE	Min...	Máx...
Driver1			0	0	0	0										
SensorA			1	3	0	1	1000		<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	0	1000		0	1000
AlarmeA			1	3	0	2	1000		<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	0	1000		0	1000
SensorB			1	3	0	3	1000		<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	0	1000		0	1000
AlarmeB			1	3	0	4	1000		<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	0	1000		0	1000
SensorC			1	3	0	5	1000		<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	0	1000		0	1000
AlarmeC			1	3	0	6	1000		<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	0	1000		0	1000
SensorD			1	3	0	7	1000		<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	0	1000		0	1000
AlarmeD			1	3	0	8	1000		<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	0	1000		0	1000
SensorE			1	3	0	9	1000		<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	0	1000		0	1000
AlarmeE			1	3	0	10	1000		<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	0	1000		0	1000

Fonte: Autor, 2019.

Após a configuração do Drive é feito o teste para garantir que a conexão foi bem-sucedida, o teste é ilustrado na Figura 16.

Figura 16 - Teste de Comunicação do Drive

Nome	Disp...	Item	P1/N...	P2/N...	P3/N...	P4/N...	Ta...	Va...	Valor	Quali...	Estampa de tempo	Valor (sem escala)
Driver1			0	0	0	0						
SensorA			1	3	0	1	1000	9	102	9	192/08/2019 11:02:25,179	102
AlarmeA			1	3	0	2	1000	9	0	9	192/08/2019 11:02:25,179	0
SensorB			1	3	0	3	1000	9	104	9	192/08/2019 11:02:25,179	104
AlarmeB			1	3	0	4	1000	9	0	9	192/08/2019 11:02:25,179	0
SensorC			1	3	0	5	1000	9	14	9	192/08/2019 11:02:25,179	14
AlarmeC			1	3	0	6	1000	9	1	9	192/08/2019 11:02:25,179	1
SensorD			1	3	0	7	1000	9	13	9	192/08/2019 11:02:25,179	13
AlarmeD			1	3	0	8	1000	9	1	9	192/08/2019 11:02:25,179	1
SensorE			1	3	0	9	1000	9	106	9	192/08/2019 11:02:26,178	106
AlarmeE			1	3	0	10	1000	9	0	9	192/08/2019 11:02:25,179	0

Fonte: Autor, 2019.

4.2 Protótipo Desenvolvido

Com a conexão do Drive pode ser realizada a construção das Telas de Supervisão e dos Relatórios que são funcionalidades de aplicabilidade do sistema de supervisão, que serão descritas nos tópicos sucessores.

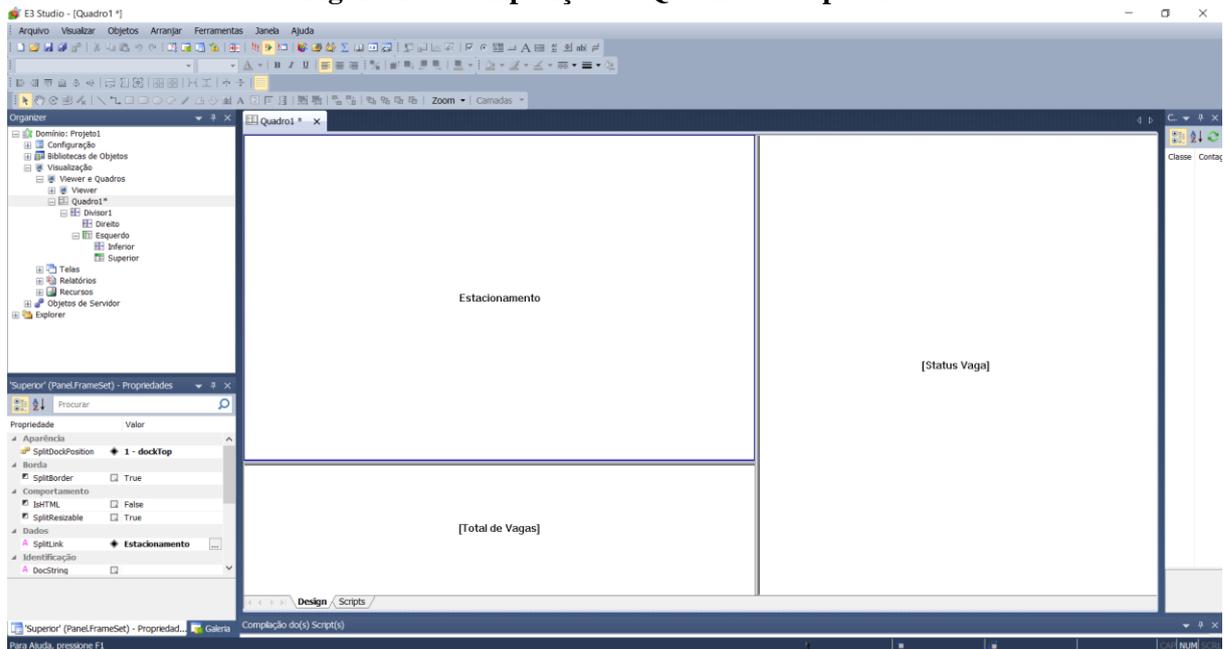
4.2.1 Telas de Supervisão

No ambiente do Eclipse E3, mostrado na Figura 8, podemos criar telas que reagem, por eventos, às ocorrências do sistema a elas configuradas. Eventos são notificações produzidas por um objeto devido a uma resposta a alguma ocorrência, como uma mudança de valor da variável (Elipse, 2012). No sistema descrito nesse trabalho precisa-se de uma tela de supervisão para o estacionamento e uma tela de relatórios onde serão dispostas as informações coletadas por as vagas.

O propósito da tela é mostrar os eventos em suas ocorrências de acordo com as informações enviadas pelos sensores. No sistema de estacionamento podemos assumir que possuímos 2 (duas) ocorrências: Disponível e Ocupado, elas serão representadas por 0 e 1 respectivamente no supervisório, de acordo com a função “*alerta*” no Anexo B.

A tela principal do sistema de supervisão de estacionamentos é composta por um Quadro, que pode ser formado por várias telas, como mostrado na Figura 17. O quadro funciona como um aglomerado de telas onde elas são organizadas, no intuito de facilitar a edição. No quadro os espaços são nomeados conforme a indicação da tela que assumirá determinada posição.

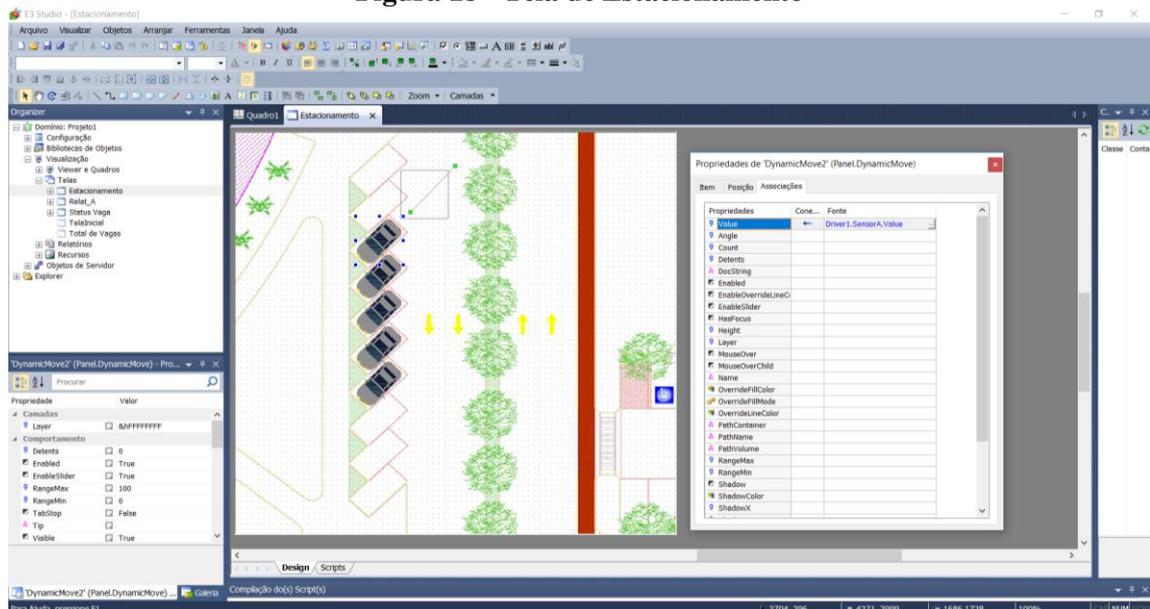
Figura 17 - Composição de Quadro de Supervisão



Fonte: Autor, 2019.

Ao distribuir as telas de supervisão pelo quadro principal, pode ser configuradas as telas que assumirão as posições. A primeira tela que irá compor o Quadro é a tela do estacionamento. A tela do Estacionamento, ilustrada na Figura 18, representa esquematicamente o estacionamento utilizado para a supervisão do modelo. Nesta tela são utilizados elementos de transição que se movem a uma ocorrência de evento da variável SensorA, sendo utilizado um elemento para cada sensor.

Figura 18 - Tela de Estacionamento

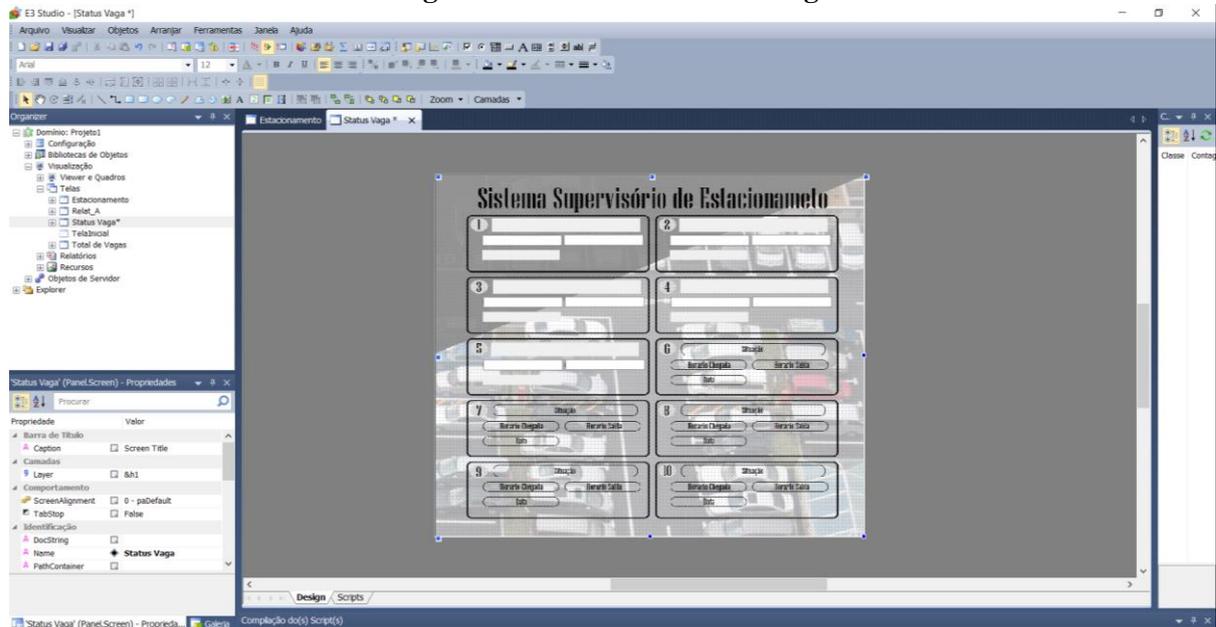


Fonte: Autor, 2019.

A segunda tela de supervisão mostra as informações referente a vaga: Status da Vaga, horário de Chegada, Horário de Saída e a Data. Ilustrada na Figura 19, a segunda tela de

supervisão contém elementos textuais e gráficos que reagem a determinado evento da variável a ele associada.

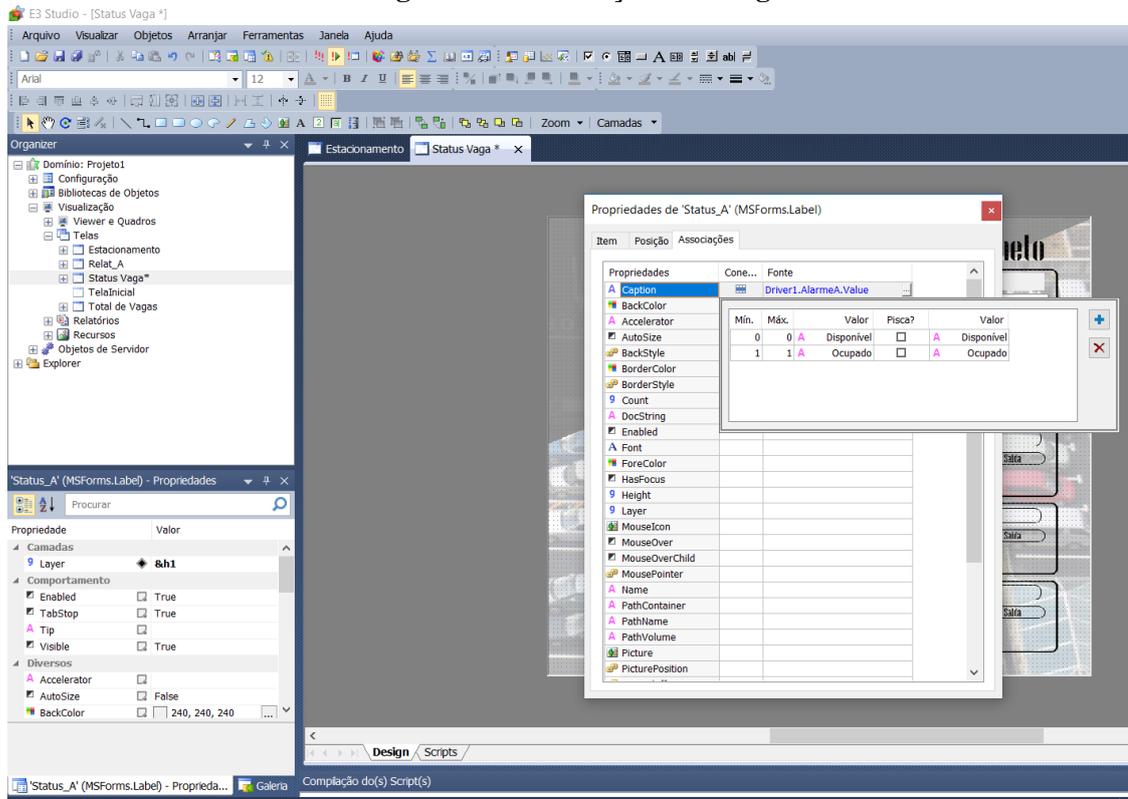
Figura 19 - Tela de Status da Vaga



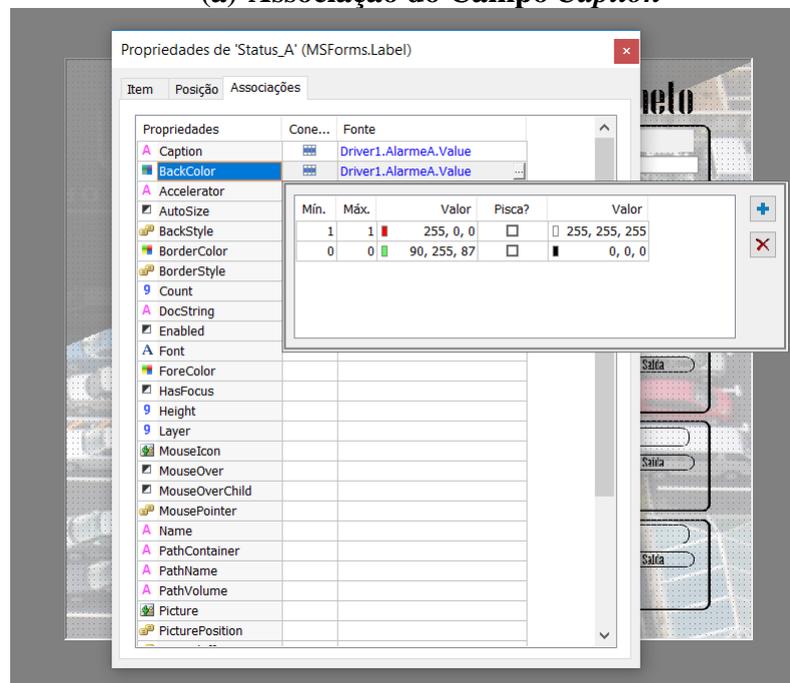
Fonte: Autor, 2019.

O campo “Situação” na “Tela de Status Vaga” é responsável por informar a situação da vaga: Disponível ou Ocupado, usando de uma simbologia de cores para representar melhor esse estado. Ao acessar a propriedade do campo de texto, o seu “Caption” pode ser configurado para reagir ao evento da Tag. A Tag escolhida para representar o evento de status da vaga é a *AlarmeLetra_da_vaga*. Acessando a tag que recebe os valores da variável que o sensor envia estabelece uma regra de associação, onde para determinado valor dessa Tag o campo se comportará de tal forma desejada, esse processo é demonstrado na Figura 20. No caso, ao valor “0” na Tag o campo mudará seu “Caption” para Disponível e seu “BackColor” será verde, caso o valor da Tag for “1” o campo mudará seu “Caption” para Ocupado e seu “BackColor” será vermelho.

Figura 20 - Associações das Tag



(a) Associação do Campo *Caption*



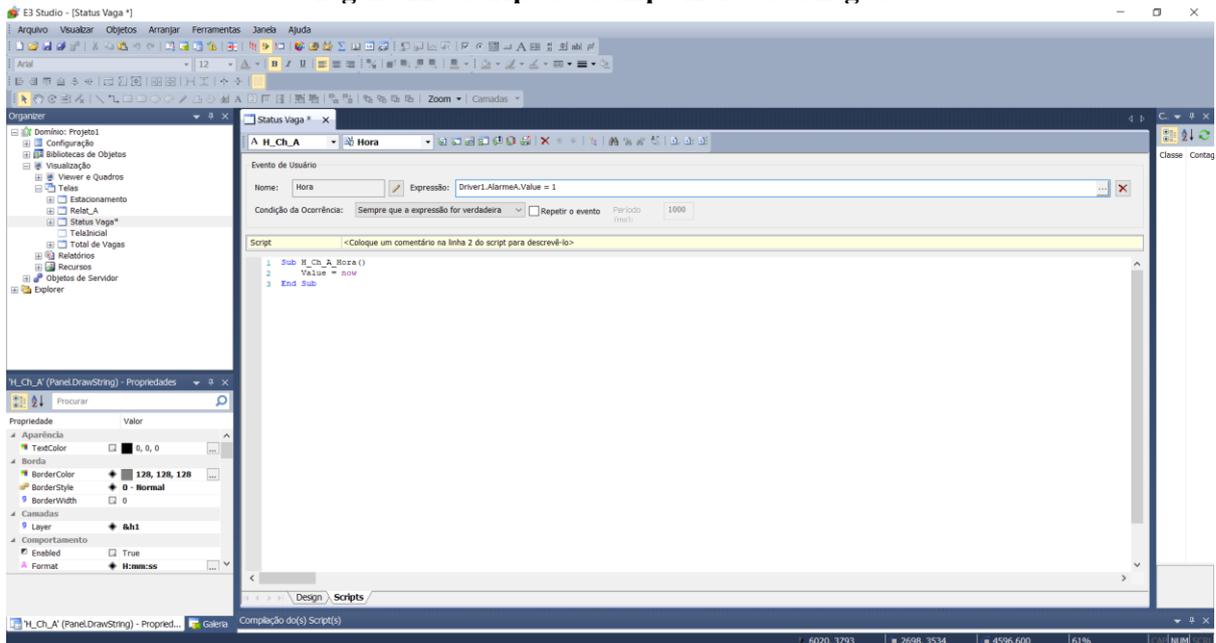
(b) Associação do Campo *BackColor*

Fonte: Autor, 2019.

Outra funcionalidade utilizada pelo campo de Status são os “scripts” eles serão comentados na parte dos relatórios, dado que esse tópico aborda apenas a parte gráfica do sistema e as suas reações aos eventos ocorrentes.

Os campos de “*Hora de Chegada*”, “*Hora de Saída*” e “*Data*”, utilizam “*scripts*” muito simples. Onde esses campos respondem a eventos com o tipo *Date* das informações. Como ilustrado pela Figura 21 o “*script*” dos campos reagem, executam o *script*, apenas a um evento “*Hora*”, que ocorre quando a expressão “*Driver1.AlarmeA.Value = 1*” for verdadeira, retornando ao valor de *Value*, a hora e a data no exato momento. Esse processo também acontece para os campos “*Hora de Saída*” e “*Data*”.

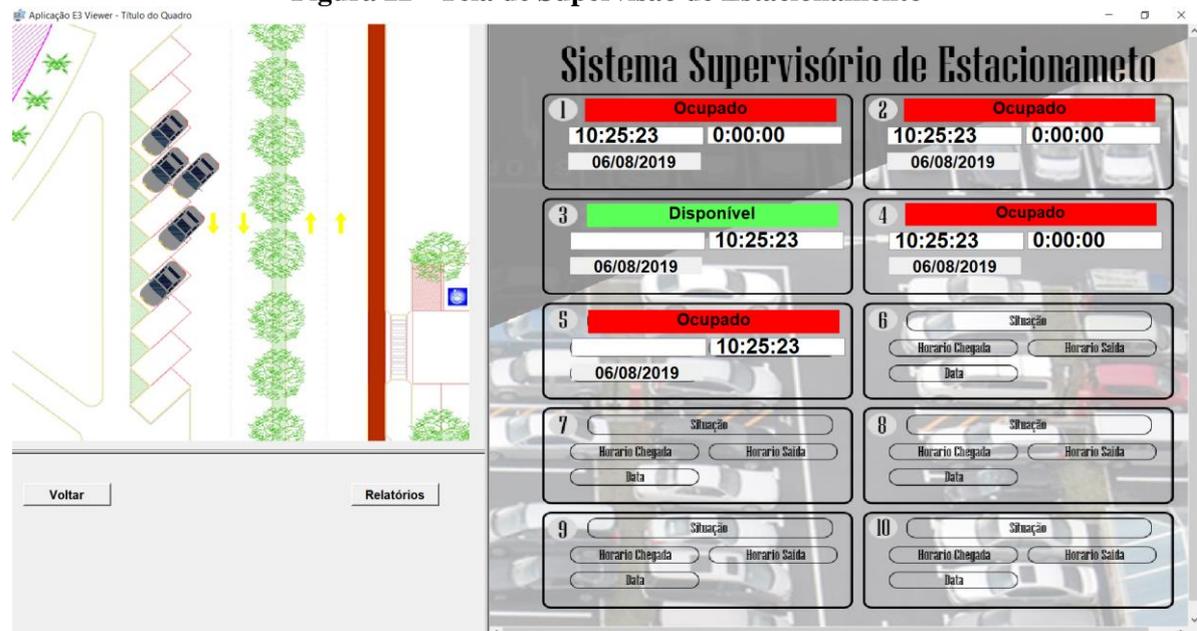
Figura 21 - Script do Campo Hora de Chegada



Fonte: Autor, 2019.

Todos esses processos de configurações de campos são realizados para os 5 (cinco) campos que representam as 5 (cinco) vagas utilizadas para a prototipação do modelo. A configuração desses campos junto com a conexão estabelecida podemos visualizar na tela de supervisão, ilustrado na Figura 22. O próximo passo será a descrição do Banco de Dados que dará suporte aos relatórios do sistema.

Figura 22 - Tela de Supervisão de Estacionamento

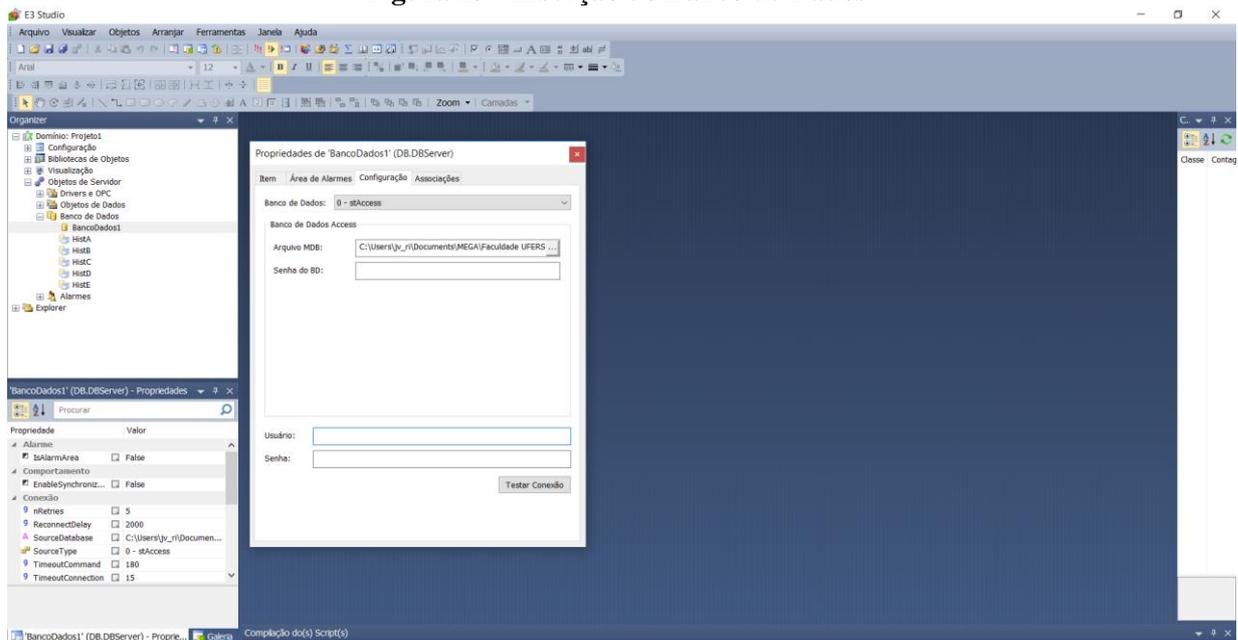


Fonte: Autor, 2019.

4.2.2 Relatório

Para a criação dos Relatórios da Supervisão é preciso a criação de campos que auxiliarão na obtenção dos dados e na escrita no banco. O Banco de Dados (BD) utilizado foi construído na ferramenta ACCESS do pacote Office 2016. Criando o BD adicionando-o no campo *Banco de Dados* no *Organize* do E3, ilustrado na Figura 23, acessando as propriedades do BD seleciona o tipo do BD e o caminho da pasta onde ele está salvo no Computador.

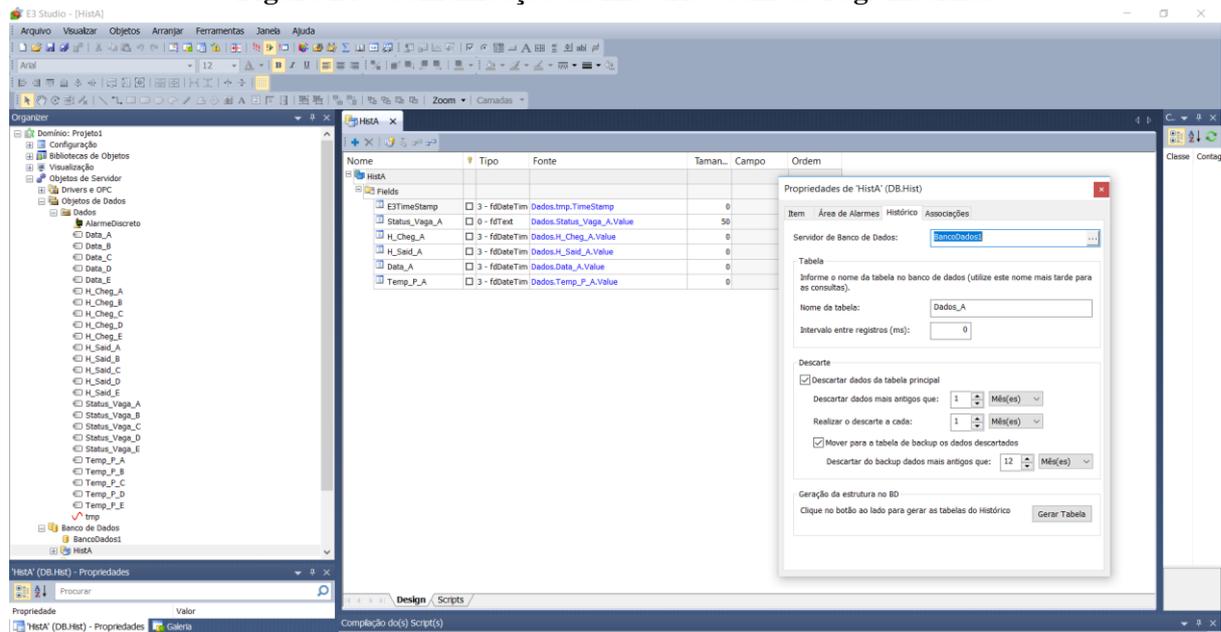
Figura 23 - Inserção do Banco de Dados



Fonte: Autor, 2019.

Após o teste da conexão, deve ser criado outro elemento importante na geração dos relatórios: o *histórico*. O *histórico* será responsável pela alimentação do BD. Para a criação do *Histórico*, realiza o mesmo processo do BD. A criação do *Histórico* implica na criação de Tags Internas que serão as responsáveis pelo armazenamento dos valores que alimentaram o histórico e conseqüentemente o BD, demonstrado na Figura 24.

Figura 24 - Comunicação do Histórico com as Tags Internas

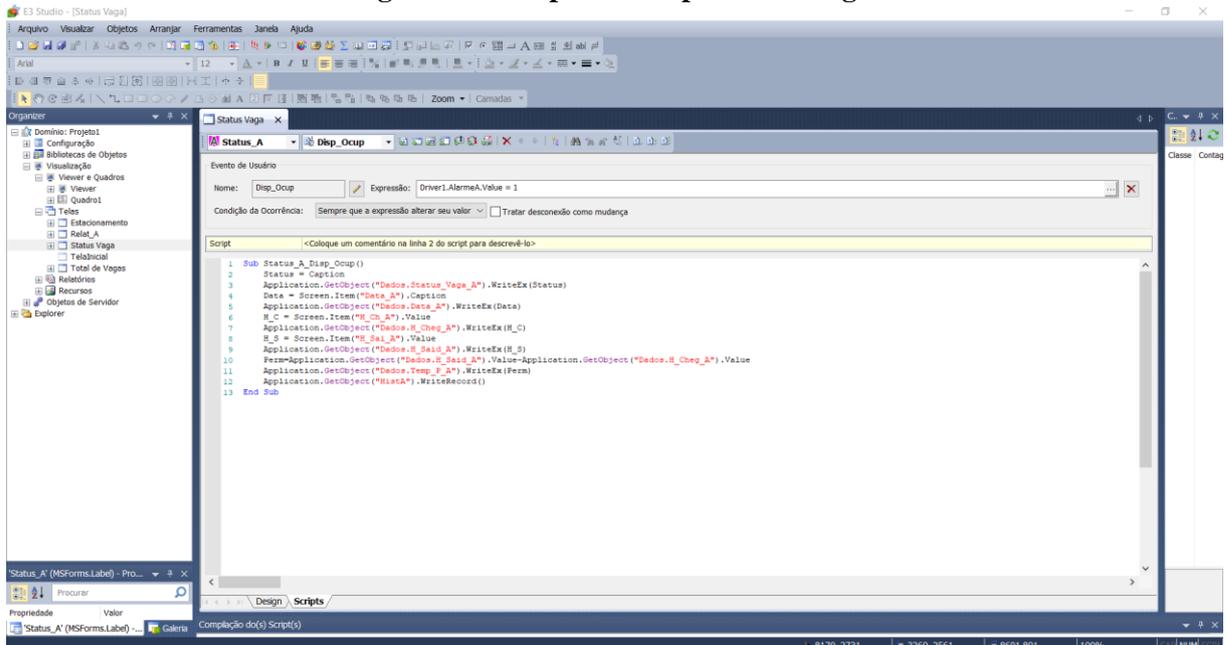


Fonte: Autor, 2019.

De acordo com a Figura 24, o *Histórico* possui atributos que possuem Tipo e fonte. Cada campo do *Histórico* é conectada à uma tag interna que o atualizará e seu tipo precisa ser compatível o que está sendo enviado. Acessando as propriedades do *Histórico* configura o BD a qual aquele histórico irá alimentar, inclusive qual tabela também será utilizada, entre outras configurações de armazenamento. Uma configuração importante é o intervalo de registros, onde é dado o tempo em que os dados são alimentados no BD, mas caso for 0, o histórico precisa de um evento para gravar os dados no BD. A criação de históricos repete-se para cada vaga existente no sistema, dado que é necessário que seja armazenado informações de cada vaga.

A alimentação do BD requer um evento, dado que um operador não teria a condição de ficar alimentando manualmente, portanto foi utilizado um evento no *script* do campo de *Status Vaga* na *Tela de Status Vaga* para enviar os dados necessários e ainda assim a requisição que o *histórico* precisa para alimentar o BD, exposta na Figura 25.

Figura 25 - Script do Campo Status Vaga.



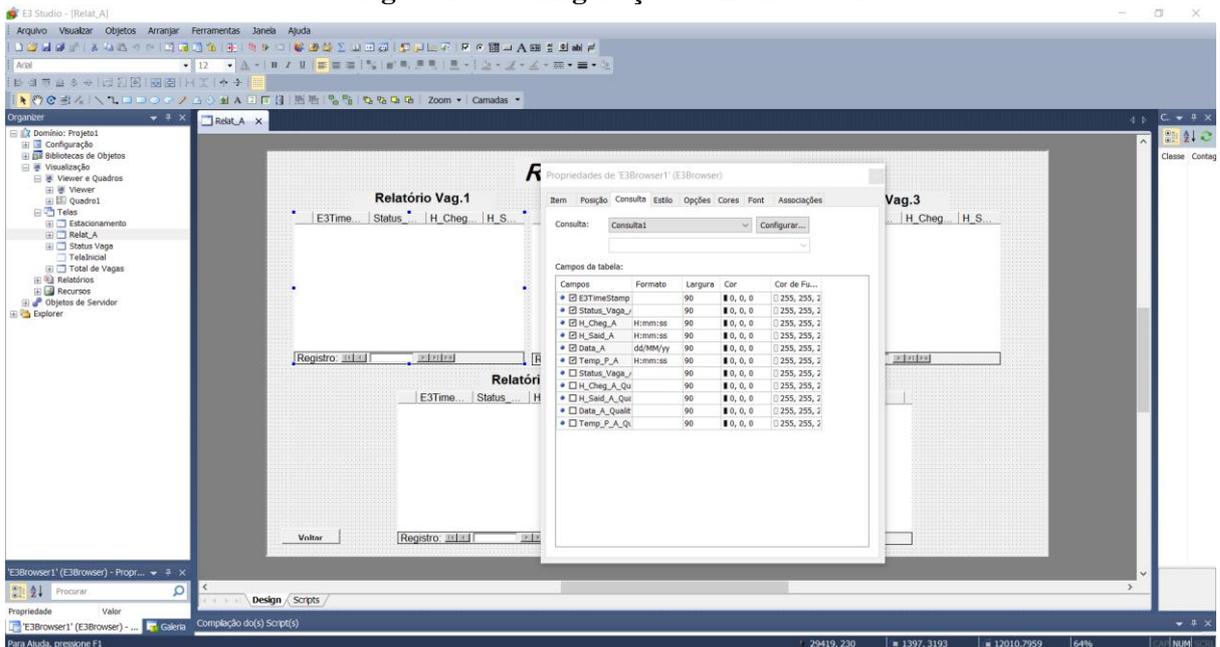
Fonte: Autor, 2019.

Os *Scripts* são módulos de Linguagem de programação que criam “procedimentos associados e eventos específicos”, garantindo flexibilidade no desenvolvimento das aplicações (Elipse, 2012). Como exposto pela figura anterior, o evento é acionado quando o valor da tag *AlarmeA* muda de valor. Recuperando os valores dos campos da tela e associando esses valores as tag internas, pelo método *Application.GetObject("Dados.Status_Vaga_A").WriteEx(Status)*, onde escreve na tag o valor, por exemplo *Status*. Ao final, dentro do *script*, é chamado um método *Application.GetObject("HistA").WriteRecord()* que atualiza as informações no *Histórico A* e alimenta o BD. Toda as configurações realizadas no *Status_Vaga_A* é repetida para os outros campos da Tela.

Agora que o BD foi configurado, os históricos e *Scripts* de atualização foram configurados pode ser disponibilizados os dados requisitados em forma de relatórios. Para isso é utilizado a ferramenta *E3Browser*. Essa ferramenta disponibiliza a disposição de informações em tabelas e pode ser configurada para dispor as informações armazenada no BD do sistema de supervisão de estacionamentos.

Em uma nova tela adiciona o elemento *E3Browser*, abre as propriedades. Nela irá conter todas as colunas (atributos) do BD que for feita referência. No caso do sistema, a primeira tabela refere-se a tabela *Dados_A*, ilustrado na Figura 26.

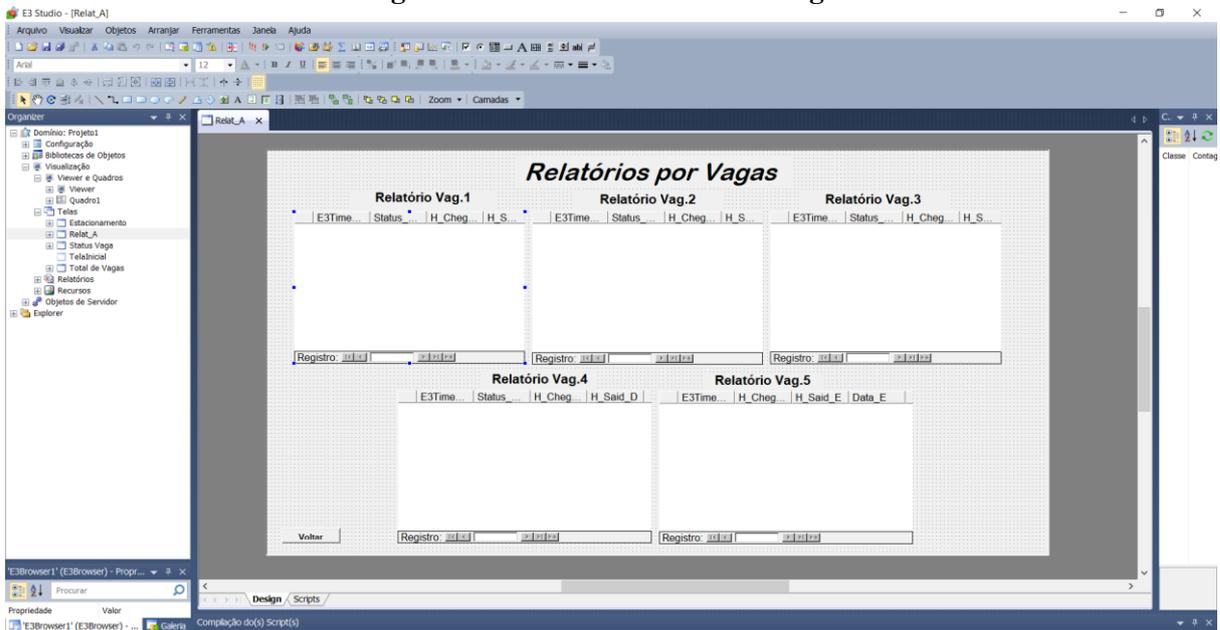
Figura 26 - Configuração do E3 Browser



Fonte: Autor, 2019.

Ao acessar as configurações do *E3Browser*, poderá ser adicionado outras tabelas ou outros atributos que ainda não compõem a tabela atual. Ao serem realizadas todas as configurações necessárias temos uma tela de relatório como demonstrada na Figura 27.

Figura 27 - Tela de Relatório de Vagas.



Fonte: Autor, 2019.

Após as configurações dos sistemas realizou-se os testes *in loco*, no estacionamento da instituição, ilustrado na Figura 28. Para implementação *in loco* foi necessário configurar extensões para a localização dos sensores nos locais adequados para a requisição dos sensores.

Figura 28 - Local de Testes *in loco*



Fonte: Autor, 2019.

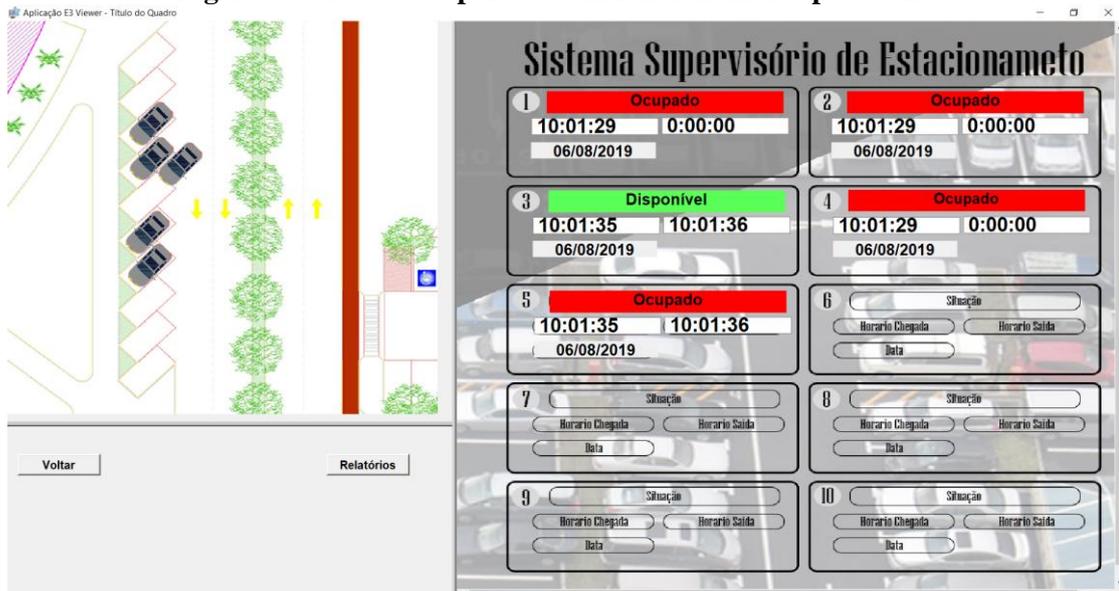
Ao iniciar os testes podemos constatar a resposta do supervisor às ocorrências que vem a acontecer, como a presença/ausência do veículo na vaga. A execução do sistema SCADA para o gerenciamento de estacionamentos é ilustrado nas Figuras 29, 30 e 31, na qual podemos observar a *Tela Inicial*, a *Tela de Supervisão*, com um evento respondendo à ocorrência do sistema de sensoriamento, e a *Tela de Relatórios*.

Figura 29 - Tela Inicial do Supervisor

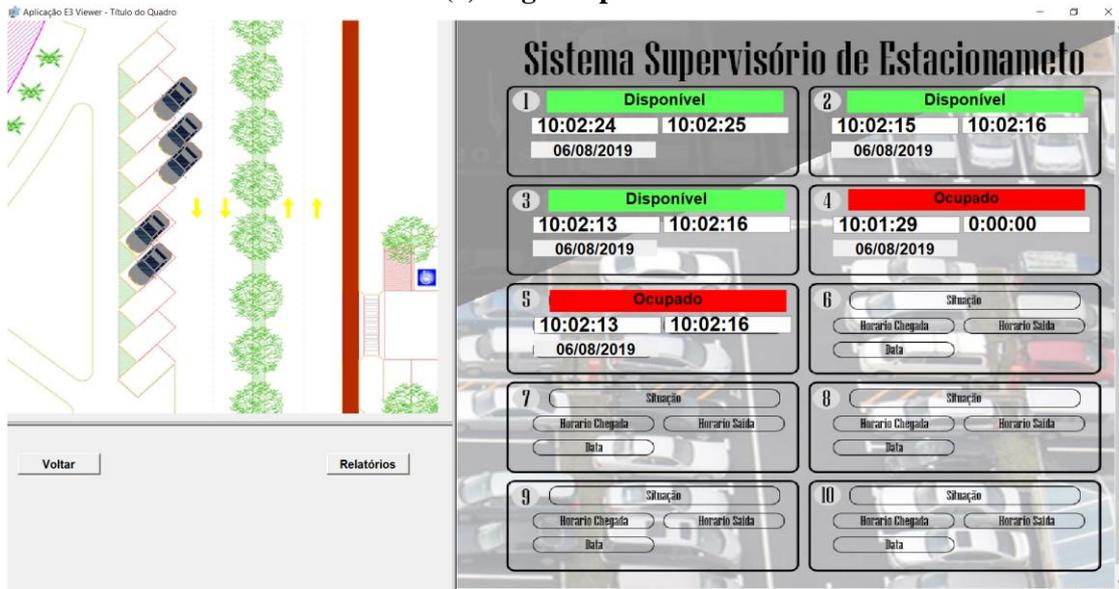


Fonte: Autor, 2019.

Figura 30 - Tela de Supervisão em funcionamento por eventos



(a) Vaga Disponível



(b) Vaga Ocupado

Fonte: Autor, 2019.

Figura 31 - Tela de Relatórios com dados do Supervisorio

Aplicação E3 Viewer - Screen Title

Relatórios por Vagas

Relatório Vag.1

Status	H. Cheg	H. Saíd	Data	Temp. P
Ocupado	10.20.48	10.21.10	05/08/19	0.00.22
Disponível	11.32.40	0.00.00	05/08/19	11.32.40
Ocupado	11.32.40	11.58.19	05/08/19	0.25.39
Disponível	9.59.10	0.00.00	06/08/19	9.59.10
Ocupado	9.59.10	9.59.56	06/08/19	0.00.46
Disponível	9.59.58	0.00.00	06/08/19	9.59.58
Ocupado	9.59.58	10.00.01	06/08/19	0.00.03
Ocupado	10.00.05	10.00.06	06/08/19	0.00.01
Disponível	10.01.29	0.00.00	06/08/19	10.01.29
Ocupado	10.01.29	10.02.13	06/08/19	0.00.44
Disponível	10.02.15	0.00.00	06/08/19	10.02.15
Ocupado	10.02.15	10.02.16	06/08/19	0.00.01
Disponível	10.02.24	0.00.00	06/08/19	10.02.24
Ocupado	10.02.24	10.02.25	06/08/19	0.00.01

Registro: 1 | 1 | 135 de 135

Relatório Vag.2

Status	H. Cheg	H. Saíd	Data	Temp. P
Ocupado	12.02.19	12.02.29	05/08/19	00.00.10
Disponível	12.02.32	0.00.00	05/08/19	25.05.17
Ocupado	12.02.32	12.02.39	05/08/19	00.00.07
Disponível	12.02.43	0.00.00	05/08/19	25.05.17
Disponível	9.59.10	0.00.00	06/08/19	24.05.17
Ocupado	9.59.10	9.59.56	06/08/19	00.00.46
Disponível	9.59.57	0.00.00	06/08/19	24.05.17
Ocupado	9.59.57	9.59.58	06/08/19	00.00.01
Disponível	10.00.02	0.00.00	06/08/19	24.05.17
Ocupado	10.00.02	10.00.03	06/08/19	00.00.01
Disponível	10.01.29	0.00.00	06/08/19	24.05.17
Ocupado	10.01.29	10.02.14	06/08/19	00.00.45
Disponível	10.02.15	0.00.00	06/08/19	24.05.17
Ocupado	10.02.15	10.02.16	06/08/19	00.00.01

Registro: 1 | 1 | 663 de 663

Relatório Vag.3

Status	H. Cheg	H. Saíd	Data	Temp. P
Ocupado	11.01.25	11.02.29	05/08/19	0.01.04
Disponível	11.02.30	0.00.00	05/08/19	11.02.30
Ocupado	11.02.30	11.03.08	05/08/19	0.00.38
Disponível	11.03.10	0.00.00	05/08/19	11.03.10
Ocupado	11.03.10	11.03.14	05/08/19	0.00.04
Disponível	11.08.15	0.00.00	05/08/19	11.08.15
Ocupado	11.08.15	11.08.24	05/08/19	0.00.09
Disponível	10.01.35	0.00.00	06/08/19	10.01.35
Ocupado	10.01.35	10.01.36	06/08/19	0.00.01
Disponível	10.02.13	0.00.00	06/08/19	10.02.13
Ocupado	10.02.13	10.02.16	06/08/19	0.00.03
Disponível	10.10.37	0.00.00	06/08/19	10.10.37
Ocupado	10.10.37	10.10.38	06/08/19	0.00.01
Disponível	10.10.52	0.00.00	06/08/19	10.10.52

Registro: 1 | 1 | 28 de 28

Relatório Vag.4

E3Time	Status	H. Cheg	H. Saíd	Data	Temp. P
05/08/20	Ocupado	11.41.37	11.41.38	05/08/19	0.00.01
05/08/20	Disponível	11.42.18	0.00.00	05/08/19	11.42.18
05/08/20	Ocupado	11.42.18	11.42.19	05/08/19	0.00.01
05/08/20	Disponível	11.42.26	0.00.00	05/08/19	11.42.26
05/08/20	Ocupado	11.42.26	11.42.27	05/08/19	0.00.01
05/08/20	Disponível	11.42.51	0.00.00	05/08/19	11.42.51
05/08/20	Ocupado	11.42.51	11.42.52	05/08/19	0.00.01
05/08/20	Disponível	11.43.58	0.00.00	05/08/19	11.43.58
05/08/20	Ocupado	11.43.58	11.43.59	05/08/19	0.00.01
05/08/20	Disponível	11.44.10	0.00.00	05/08/19	11.44.10
05/08/20	Ocupado	11.44.10	11.44.11	05/08/19	0.00.01
05/08/20	Disponível	11.45.34	0.00.00	05/08/19	11.45.34
05/08/20	Ocupado	11.45.34	11.51.40	05/08/19	0.06.06
05/08/20	Disponível	11.53.27	0.00.00	05/08/19	11.53.27
05/08/20	Ocupado	11.53.27	11.53.28	05/08/19	0.00.01

Registro: 1 | 1 | 324 de 324

Relatório Vag.5

E3Time	Status	H. Cheg	H. Saíd	Data	Temp. P
05/08/20	Disponível	11.16.11	0.00.00	05/08/19	11.16.11
05/08/20	Ocupado	11.30.03	0.00.00	05/08/19	11.30.03
05/08/20	Disponível	11.32.40	0.00.00	05/08/19	11.32.40
06/08/20	Disponível	10.01.35	10.01.36	06/08/19	0.00.01
06/08/20	Ocupado	10.01.35	10.01.36	06/08/19	0.00.01
06/08/20	Disponível	10.02.13	10.02.16	06/08/19	0.00.03
06/08/20	Ocupado	10.02.13	10.02.16	06/08/19	0.00.03
06/08/20	Disponível	10.02.13	10.02.16	06/08/19	0.00.03
06/08/20	Ocupado	10.02.13	10.02.16	06/08/19	0.00.03
06/08/20	Disponível	10.02.13	10.02.16	06/08/19	0.00.03
06/08/20	Ocupado	10.02.13	10.02.16	06/08/19	0.00.03
06/08/20	Disponível	10.10.37	10.10.38	06/08/19	0.00.01
06/08/20	Ocupado	10.10.52	10.10.53	06/08/19	0.00.01

Registro: 1 | 1 | 19 de 19

[Voltar](#)

Fonte: Autor, 2019.

5. CONSIDERAÇÕES FINAIS

A inclusão da automação nos processos do cotiando já é uma prática bastante comum em todos os âmbitos profissionais, a resolução de problemas e/ou prover facilidade de realização dessas atividades torna-se indispensável em uma sociedade transitória da evolução tecnológica. O seguinte trabalho apresenta uma solução para gerenciar estacionamentos automatizados a partir das técnicas *SCADA*.

A ferramenta desenvolvida provê aos operadores uma melhor administração sobre os recursos de cada vaga de estacionamento, promovendo um maior controle de dados à informação disponibilizada pelo sistema. Todos eventos ressaltados no supervisório apontam as mudanças no estado dos sensores, informando o estado atual da vaga. Além disso esses dados dispostos na tela de supervisão estão sendo armazenados e organizados em tabelas, por vagas, em um banco de dado. O banco de dados gerados com a aplicação proporciona o armazenamento dos dados e realizar o tratamento desses.

Na implementação desse sistema ocorreram algumas dificuldades que vieram a somar ao resultado do trabalho, como a utilização de uma versão mais atualizada do software de supervisão da Elipse dado que o protocolo MODBUS estava com erros na comunicação devido a sua versão. Um outro problema que foi observado é o uso de cabos para o projeto, que aumentam os riscos à falhas, além do envio dos dados pelo Arduino não ser em tempo hábil de resposta. Outro problema que venha a ser destacada é a utilização do sistema de sensoriamento, pois qualquer objeto pode ser identificado com o sensor, dificultado a veracidade da informação. Por exemplo, qualquer objeto ou animal pode vir a ocupar a vaga danificando a informação do sensor.

Como trabalhos futuros, foram detectadas algumas sugestões de melhorias e aplicações que venham a agregar na qualidade dos serviços disponibilizados com a inserção de ferramentas que utilizem as informações trabalhadas pelo sistema. Como ferramenta a serem agregadas ao trabalho, pode ser mencionada a comunicação *Wireless* entre os componentes e o armazenamento das informações no conceito de nuvem. Uma outra sugestão que venha a melhorar a usabilidade do sistema é a inserção de técnicas de IoT, disponibilizando a plataforma em aplicações portáteis.

REFERÊNCIAS

ARDUINO. **O que é Arduino?**.2019. Disponível em: <https://www.arduino.cc>. Acesso em: 1 ago. 2019.

BANDEIRA, Thayanne Barros et *al.* **PROTÓTIPO DE ESTACIONAMENTO AUTOMATIZADO UTILIZANDO MODELO COMPUTACIONAL MATRICIAL E MICROCONTROLADOR ARDUINO**. In: Congresso Nacional de Matemática Aplicada à Indústria, 2014, Calda Novas – Goiás. *Anais...* Calda Novas – Goiás: CNMAI, 2014.

BARROS, Thiago Medeiros. **UMA COLUNA DE DESTILAÇÃO SIMULADA INTEGRADA A INSTRUMENTAÇÃO REAL** – Natal. 2012. 75f. Dissertação de Mestrado em Engenharia Elétrica, Universidade Federal do Rio Grande do Norte. Natal, 2012.

BAYER, Fernando Mariano et *al.* **AUTOMAÇÃO DE SISTEMAS**, 4. ed. – Santa Maria: Universidade Federal de Santa Maria, Colégio Técnico Industrial de Santa Maria; Escola Técnica Aberta do Brasil, 2011.

BRASIL. DENATRAN. **RELATÓRIO DA FROTA DE VEÍCULOS – 2019** – Brasília, 2019.

BRASIL. INSTITUTO BRASILEIRO DE ADMINISTRAÇÃO MUNICIPAL – IBAM. **MOBILIDADE E POLÍTICA URBANA: SUBSÍDIOS PARA UMA GESTÃO INTEGRADA**. Coordenação de Lia Bergman e Nidia Inês Albesa de Rabi. 52 p. – Rio de Janeiro: IBAM; Ministério das Cidades, 2005.

BRASIL. INSTITUTO DE PESQUISA ECONÔMICA APLICADA – IPEA. **TEXTO PARA DISCUSSÃO - MOBILIDADE URBANA SUSTENTÁVEL: CONCEITOS TENDÊNCIAS E REFLEXÕES** – Rio de Janeiro, 2016.

BRASIL. MINISTÉRIO DAS CIDADES. **MOBILIDADE URBANA É DESENVOLVIMENTO URBANO**, 39 p. – Brasília, 2005.

CASSIANO, Luciano V. **ESTUDO DE IMPLANTAÇÃO DE UM SISTEMA DE ESTACIONAMENTO ROTATIVO NA REGIÃO CENTRAL DA CIDADE DE CAMPO MOURÃO-PR- CAMPO MOURÃO**. 2014. 43f. Trabalho de Conclusão de Curso (Graduação) – Engenharia Civil, Universidade Tecnológica Federal do Paraná. Campo Mourão, 2014.

CAVAMURA, Humberto Fernando Massaharu e MITSUHASHI, MARLOS KENJY. **SISTEMA DE GERÊNCIA DE VAGAS DE ESTACIONAMENTO** – Curitiba. 2014. 108f. Trabalho de Conclusão de Curso (Graduação) – Engenharia Elétrica. Universidade Tecnológica Federal do Paraná. Curitiba, 2014.

CORREA, Tiago. **MONITORAMENTO DE UM SISTEMA DE EMERGÊNCIA INDUSTRIAL, APLICANDO O SOFTWARE SUPERVISÓRIO SCADA (ELIPSE)** – Itatiba. 2012. 55f. Trabalho de Conclusão de Curso (Graduação) – Engenharia Elétrica. Universidade São Francisco. Itatiba, 2012.

ELIPSE Knowledgebase. **COMO O ELIPSE E3 FUNCIONA**. 2019. Disponível em: <https://kb.elipse.com.br/como-o-elipse-e3-funciona/>. Acesso em: 29 jul. 2019.

ELIPSE.MANUAL DE REFERÊNCIA DE SCRIPTS DO E3. Versão 4.0, 2012

FRANÇA, Tiago Cardoso. **PROJETO DE UM SISTEMA SUPERVISÓRIO PARA UMA PLANTA MECATRÔNICA DE ESTOCAGEM DE PEÇAS** – Rio de Janeiro. 2015. 197f. Trabalho de Conclusão de Curso (Graduação) – Engenharia Elétrica. Universidade Federal do Rio de Janeiro. Rio de Janeiro, 2015.

FUJITA, Henrique de Souza et al. **VAGA SIMPLES: ESTACIONAMENTO AUTOMATIZADO** – Criciúma. 2016. 38f. Relatório apresentado à disciplina de Práticas Profissionais e Sistemas Digitais II. Criciúma, 2016.

INDUCTIVE AUTOMATION. **O QUE É O SCADA?** 2018. Disponível em: <<https://inductiveautomation.com/resources/article/what-is-scada>>. Acesso em: 04 jul. 2019.

JUNIOR, Antônio Carlos Buch e GRUNEWALDER, Marcos. **DESENVOLVIMENTO DE UM CONCENTRADOR DE DADOS VIA SOFTWARE ELIPSE SCADA PARA ENSAIO DE DESEMPENHO EM USINA TERM ELÉTRICA ESTUDO DE CASO –** Curitiba. 2013. Trabalho de Conclusão de Curso (Graduação) – Tecnologia em Eletrotécnica. Universidade Tecnológica Federal do Paraná. Curitiba, 2013.

KRETZER, Philipe e AZEVEDO, Castro de. **DESENVOLVIMENTO DE UM SISTEMA SUPERVISÓRIO E LÓGICAS DE CLP NO AMBIENTE DE GERAÇÃO DE ENERGIA.** 2013. 108f. Trabalho de Conclusão de Curso (Graduação) – Engenharia de Controle e Automação. Universidade Federal de Santa Catarina. Florianópolis, 2013.

LEANDRO, Victor Luiz dos Santos et *al.* **PROJETO VAGA FÁCIL: UMA SOLUÇÃO AUTOMATIZADA NA VIABILIZAÇÃO DE VAGAS EM ESTACIONAMENTOS PÚBLICOS E PRIVADOS.** In: Congresso Norte Nordeste de Pesquisa e Inovação, 4., 2012, Palmas – Tocantins. *Anais...* Palmas – Tocantins: CONNEPI, 2012.

PEREIRA, Carlos Eduardo e PARDI, Wilson Junior. *A SUPERVISORY TOOL FOR REAL-TIME INDUSTRIAL AUTOMATION SYSTEMS.* In *Sixth IEEE International Symposium on Object-Oriented Real-Time Distributed Computing*, 2003.

RAHMAN, Saidur e BHOUMIK, Poly. *IoT Based Smart Parking System.* *International Journal of Advances in Computer and Electronics Engineering*, Mizoram – Índia, v. 4, n. 1, p. 11–16, 2019.

REICHERT, Rogers Guilherme et *al.* **ESTACIONAMENTOS INTELIGENTES COMO SOLUÇÃO AO CONGESTIONAMENTO DE GRANDES CENTROS URBANOS.** In: Congresso Brasileiro de Educação em Engenharia, 39., 2011, Blumenau – Santa Catarina.

RIBEIRO, Marco Antônio. **FUNDAMENTOS DA AUTOMAÇÃO.** 1. ed. Salvador: Tek Treinamento & Consultoria Ltda, 2003.

ROGGIA, Leandro e FUENTES, Rodrigo Cardozo. **AUTOMAÇÃO INDUSTRIAL –** Santa Maria: Universidade Federal de Santa Maria, Colégio Técnico Industrial de Santa Maria; Escola Técnica Aberta do Brasil, 2016.

SILVA, Antonio Diermison Rosa da et *al.* **PROTOTIPAGEM DE ESTACIONAMENTO INTELIGENTE COM COMUNICAÇÃO SEM FIO.** In: Mostra Nacional de Robótica, 2013, Teresina – Piauí. *Anais...* Teresina – Piauí: MNR, 2013. p. 190-193.

SILVEIRA, Cristiano Bertulucci. **CITISYSTEMS - O QUE É AUTOMAÇÃO INDUSTRIAL?** 2013. Disponível em: <<https://www.citisystems.com.br/o-que-e-automacao-industrial/>>. Acesso em: 04 jul. 2019.

SOUZA, Rodrigo Barbosa de. **UMA ARQUITETURA PARA SISTEMAS SUPERVISÓRIOS INDUSTRIAIS E SUA APLICAÇÃO EM PROCESSOS DE ELEVAÇÃO ARTIFICIAL DE PETRÓLEO** – Natal, RN, 2005. 53p. Dissertação de Mestrado em Engenharia Elétrica, Universidade Federal do Rio Grande do Norte. Natal, 2005.

STEFANELLI, Victor; e PERO, Valéria; **A QUESTÃO DA MOBILIDADE URBANA NAS METRÓPOLES BRASILEIRAS.** Revista de Economia Contemporânea, p 366-402, 2015.

THOMSEN, Adilson. **Como conectar o Sensor Ultrassônico HC-SR04 ao Arduino.** 2011. Disponível em: <https://www.filipeflop.com/blog/sensor-ultrassonico-hc-sr04-ao-arduino/> . Acesso em: 1 ago. 2019.

VIANNA, Guilherme Szczerbacki Besserman. **MOBILIDADE URBANA NO BRASIL: UMA ESTIMATIVA DO PRODUTO PERDIDO EM TRÂNSITO** – Rio de Janeiro. 2013. 50f. Trabalho de Conclusão de Curso (Graduação) - Economia. Universidade Federal do Rio de Janeiro. Rio de Janeiro, 2013.

VICENTE, Thiago Pires Romanelli. **CONTROLE INTELIGENTE DE VAGAS PARA ESTACIONAMENTO UTILIZANDO O CONCEITO DE INTERNET DAS COISAS** – São Carlos. 2016. 92f. Trabalho de Conclusão de Curso (Graduação) – Engenharia Elétrica. Universidade de São Paulo. São Carlos, 2016.

Anexo A – Código do Sensor Ultrassônico

```
#include <Ultrasonic.h>
#define Apino_t 2
#define Apino_e 3
#define Bpino_t 5
#define Bpino_e 6
#define Cpino_t 8
#define Cpino_e 9
#define Dpino_t 11
#define Dpino_e 12
#define Epino_t 22
#define Epino_e 23
Ultrasonic ultrasonicA(Apino_t, Apino_e);
Ultrasonic ultrasonicB(Bpino_t, Bpino_e);
Ultrasonic ultrasonicC(Cpino_t, Cpino_e);
Ultrasonic ultrasonicD(Dpino_t, Dpino_e);
Ultrasonic ultrasonicE(Epino_t, Epino_e);
void setup() {

}
void loop() {
  long microsecA = ultrasonicA.timing();
  float cmMsecA = ultrasonicA.convert(microsecA, Ultrasonic::CM);
  Serial.print("Distancia em cm do A: ");
  Serial.print(cmMsecA);
  Serial.println(" ");

  delay(500);
}
```

Anexo B – Código do Protocolo Modbus

```
#include <SimpleModbusSlave.h>

#define Apino_t 2
#define Apino_e 3
#define Bpino_t 5
#define Bpino_e 6
#define Cpino_t 8
#define Cpino_e 9
#define Dpino_t 11
#define Dpino_e 12
#define Epino_t 22
#define Epino_e 23

enum {
    Sen_1,
    Alr_1,
    Sen_2,
    Alr_2,
    Sen_3,
    Alr_3,
    Sen_4,
    Alr_4,
    Sen_5,
    Alr_5,
    HOLDING_REGS_SIZE
};
unsigned int holdingRegs[HOLDING_REGS_SIZE];

void setup() {
    modbus_configure(&Serial, 9600, SERIAL_8N1, 1, 2, HOLDING_REGS_SIZE,
holdingRegs);
    modbus_update_comms(9600, SERIAL_8N1, 1);
}
void loop() {
    modbus_update();

    long microsecA = ultrasonicA.timing();
    float cmMsecA = ultrasonicA.convert(microsecA, Ultrasonic::CM);
    Serial.print("Distancia em cm do A: ");
    Serial.print(cmMsecA);
    Serial.println(" ");
    holdingRegs[Sen_1] = cmMsecA;
    holdingRegs[Alr_1] = alerta(cmMsecA);

    delay(500);
}
float alerta(float cmMsec) {
    if (cmMsec < 50.00) {
        return 1;
    } else {
        return 0;
    }
}
}
```