

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
DEPARTAMENTO ACADÊMICO DE ELETRÔNICA
CURSO SUPERIOR DE TECNOLOGIA EM SISTEMAS DE TELECOMUNICAÇÕES

CLÁUDIO MARCELO PISKA

MÁRCIO CHUGAN PEREIRA

MÁRIO DO NASCIMENTO

**TELESITE: Monitoramento e Gerência de Estações Remotas de TV
e Telecom**

TRABALHO DE CONCLUSÃO DE CURSO

CURITIBA
2018

CLÁUDIO MARCELO PISKA
MARCIO CHUGAN PEREIRA
MÁRIO DO NASCIMENTO

TELESITE: Monitoramento e Gerência de Estações Remotas de TV e Telecom

Trabalho de Conclusão de Curso de Graduação, apresentado ao Curso Superior de Tecnologia em Sistemas de Telecomunicações, do Departamento Acadêmico de Eletrônica, da Universidade Tecnológica Federal do Paraná – UTFPR, como requisito parcial para obtenção do título de Tecnólogo.

Orientadora: Prof.^a Dra. Tânia Lucia Monteiro

CURITIBA

2018

TERMO DE APROVAÇÃO

CLÁUDIO MARCELO PISKA
MARCIO CHUGAN PEREIRA
MÁRIO DO NASCIMENTO

TELESITE: Monitoramento e Gerência de Estações Remotas de TV e Telecom

Este trabalho de conclusão de curso foi apresentado no dia 04 de julho de 2018, como requisito parcial para obtenção do título de Tecnólogo em Sistemas de Telecomunicações, outorgado pela Universidade Tecnológica Federal do Paraná. Os alunos foram arguidos pela Banca Examinadora composta pelos professores abaixo assinados. Após deliberação, a Banca Examinadora considerou o trabalho aprovado.

Prof. Dra. Tânia Lucia Monteiro
Coordenadora de Curso
Departamento Acadêmico de Eletrônica

Prof. M. Sc. Sérgio Moribe
Responsável pela Atividade de Trabalho de Conclusão de Curso
Departamento Acadêmico de Eletrônica

BANCA EXAMINADORA

Prof. Dr. Kleber Kendy Horikawa Nabas
UTFPR

Prof. MSc. Alexandre Jorge Miziara
UTFPR

Prof.^a Dra. Tânia Lucia Monteiro
Orientador - UTFPR

“A Folha de Aprovação assinada encontra-se na Coordenação do Curso”

À nossas famílias que sempre estiveram ao nosso lado nos incentivando a seguir em frente nos momentos mais difíceis e compartilhando das nossas vitórias.

AGRADECIMENTOS

Agradecemos a Deus pela vida, pois Ele é o condutor mestre que garante o fluxo da corrente vital e das nossas ações. Esperamos estar no caminho certo.

Agradecemos também aos nossos mestres “Professores” e escritores que, de uma forma ou de outra, nos conduziram pelo caminho até chegarmos neste trabalho final, na verdade é um novo começo em nossas vidas profissionais.

À Sociedade Rádio Emissora Paranaense S/A – RPC Curitiba, pelo empréstimo de local e sensores para testes do projeto.

RESUMO

PISKA, Cláudio Marcelo. PEREIRA, Márcio Chugan. NASCIMENTO, Mário do. **TELESITE: Monitoramento e Gerência de Estações Remotas de TV e Telecom.** 2018. 87 f. Trabalho de Conclusão de Curso (Curso Superior de Tecnologia em Sistemas de Telecomunicações), Departamento Acadêmico de Eletrônica, Universidade Tecnológica Federal do Paraná. Curitiba, 2018.

Este trabalho tem como objetivo desenvolver uma solução de telemetria com parâmetros configuráveis, capaz de monitorar remotamente um posto de retransmissão de Televisão ou Telecomunicações. Esta necessidade partiu da percepção técnica em campo, onde as falhas geralmente são detectadas quando não é mais possível evitar a perda de qualidade de serviço ou mesmo afetar sua disponibilidade. Além disso, muitas vezes não se pode precisar o que exatamente ocorreu. Com estas necessidades, encontrou-se uma possibilidade de solução razoavelmente simples e de baixo custo, baseada na placa de desenvolvimento *Raspberry Pi*, que possui muitos recursos e possibilitam captar dados de sensores, tratá-los e fazer interações com a central de gerência, tudo isso utilizando softwares livres. Com este sistema de telemetria busca-se ter em tempo real informações de funcionamento de uma estação remota, prevenção, informação de falhas e análise de dados, itens importantes para tomada de decisão mais assertiva. O produto pode ser uma importante ferramenta para ajudar a diminuir altos custos com deslocamentos de mão de obra técnica e troca de equipamentos.

Palavras chave: Monitoração. Telemetria. Raspberry.

ABSTRACT

PISKA, Cláudio Marcelo. PEREIRA, Márcio Chugan. NASCIMENTO, Mário do. **TELESITE: Monitoring and Management of Remote TV and Telecom Stations.** 2018. 87 f. Trabalho de Conclusão de Curso (Curso Superior de Tecnologia em Sistemas de Telecomunicações), Departamento Acadêmico de Eletrônica, Universidade Tecnológica Federal do Paraná. Curitiba, 2018.

This work aims to develop a telemetry solution with configurable parameters, capable of remotely monitoring a television or telecommunications relay station. This need arose from the technical perception in the field, where failures are usually detected when it is no longer possible to avoid the loss of quality of service or even affect its availability. In addition, it is often not possible to determine what exactly occurred. With these needs, a reasonably simple and low-cost solution based on the Raspberry Pi development board has been found, which has many features that enable it to capture sensor data, treat it, and make interactions with central management, all of which using free software. With this telemetry system, it is sought to have real-time information of a remote station's operation, prevention, failure information, data analysis, important items for more assertive decision making. The product can be an important tool to help reduce high costs with displacements of technical labor and equipment exchange.

Keywords: Monitoration. Telemetry. Raspberry.

LISTA DE ILUSTRAÇÕES

Figura 1 - <i>Raspberry Pi 2</i>	21
Figura 2 - Periféricos conectados ao <i>Raspberry</i>	23
Figura 3 - Exemplo de sistema de supervisão.....	30
Figura 4 - GPIO (Pinos programáveis de entrada/saída)	31
Figura 5 - Diagrama em blocos	34
Figura 6 - Circuito de Leituras Analógicas.....	36
Figura 7 - Cabo de interligação	37
Figura 8 - Placas montadas.....	38
Figura 9 - Projeto Montado.....	39
Figura 10 - Entrada de Alimentação no Telesite	39
Figura 11 - Conexões do <i>Raspberry</i>	40
Figura 12 - Tela Proposta.....	43
Figura 13 – Simulação de Sensores	45
Figura 14 - Tela do Provedor para VPN	47
Figura 15 - Tela Inicial do Telesite	48
Figura 16 - Tela de Configurações do Telesite.....	49
Figura 17 - Tela de Configuração de Entrada Analógica.....	50
Figura 18 - Tela de Configuração de Entrada Digital	51
Figura 19 - Tela de Parâmetros Analógicos do Telesite.....	51
Figura 20 - Tela de Parâmetros Digitais do Telesite	52

LISTA DE SIGLAS

3G	Terceira Geração de celulares e internet móvel
ARM	<i>Advanced RISC Machine</i>
GB	<i>GigaByte</i>
GPIO	<i>General Purpose Input/Output</i>
IP	<i>Internet Protocol</i>
HDMI	<i>High Definition Multimedia Interface</i>
PC	<i>Personal Computer</i>
PERL	<i>Practical Extraction and Report Language</i>
PHP	<i>Hypertext Preprocessor - Personal Home Page</i>
RAM	<i>Random Access Memory</i>
RISC	<i>Reduced Instruction Set Computer</i>
SGBD	Sistema de Gerenciamento de Banco de Dados
SNMP	<i>Simple Network Management Protocol</i>
SO	Sistema Operacional.
SQL	<i>Structured Query Language</i>
TV	Televisão
USB	<i>Universal Serial Bus</i>
WAMP	<i>Windows, Apache, MySQL, PHP e Perl.</i>
VPN	<i>Virtual Private Network</i>

SUMÁRIO

1	INTRODUÇÃO	11
1.1	PROBLEMA	13
1.2	JUSTIFICATIVA	13
1.3	OBJETIVOS	14
1.3.1	Geral	14
1.3.2	Objetivos específicos	14
2	METODOLOGIA	15
3	FUNDAMENTAÇÃO TEÓRICA	17
3.1	SISTEMAS DE COMUNICAÇÕES	17
3.1.1	Meios sem fio	17
3.1.2	Central de Monitoração	18
3.1.3	Telemetria	18
3.2	IOT – Internet das Coisas	20
3.3	RASPBERRY PI	21
3.3.1	Ambiente de desenvolvimento	21
3.3.2	Programação do <i>Raspberry</i>	23
3.3.3	Ferramentas Instaladas	24
3.4	SENSORES	28
3.4.1	Sensores analógicos	28
3.4.2	Sensores digitais	28
3.5	SISTEMAS DE SUPERVISÃO	29
4	DESENVOLVIMENTO	31
4.1	FUNCIONAMENTO DO DISPOSITIVO	32
4.1.1	Leituras analógicas	32
4.1.2	Leituras digitais	33
4.1.3	Telecomandos	33
4.1.4	Protocolo SNMP	33
4.2	<i>HARDWARE</i>	34
4.2.1	Componentes do <i>hardware</i>	34
4.2.2	Tratativas durante a elaboração e testes	35
4.2.3	Montagem	37

4.3	SOFTWARE	41
4.3.1	Código desenvolvido para as leituras de sensores	42
4.3.2	Proposta de interface.....	42
5	APRESENTAÇÃO E ANÁLISE DOS RESULTADOS.....	44
6	CONSIDERAÇÕES FINAIS.....	54
	REFERÊNCIAS.....	56
	APÊNDICE A - Código Fonte da Aplicação em Python	59
	APÊNDICE B - Código Fonte da Página Inicial.php	63
	APÊNDICE C - Código Fonte Tela configura.php.....	71
	APÊNDICE D - Código Fonte Tela altera.php.....	76
	APÊNDICE E - Código Fonte Arquivo de Estilos style.css	81
	ANEXO A - Folha de Dados do CI MCP3008	87

1 INTRODUÇÃO

A evolução tecnológica, facilmente percebida nos dias atuais propicia o tráfego de um grande volume de informações em um curto espaço de tempo, fazendo com que, o que acontece de um lado do mundo, já seja noticiado no outro.

Isso é possibilitado pelo avanço das telecomunicações, especialmente com a TV e banda larga nas redes de telefonia. Para que isso seja possível, é necessária a construção da infraestrutura de retransmissão, que demanda grandes investimentos. Mesmo após a implantação, há um gasto elevado na manutenção destes postos de retransmissão, seja pelo consumo de energia, manutenções ou mesmo por deslocamentos para verificações básicas do sistema. Há ainda perdas ocasionadas pela indisponibilidade de serviços, seja por falta de manutenção, vandalismo, ou demora na percepção de falhas.

Em tempo não muito distante, pouco mais de vinte anos, os equipamentos de transmissão e processamento de dados, contavam com uma estrutura ainda bem simples, focada essencialmente na execução de serviços de dados. A tecnologia não permitia muitas formas de monitoração dos equipamentos e quando possibilitava, nem sempre era possível fazê-lo remotamente.

Principalmente nas retransmissoras de televisão, pouco se podia monitorar remotamente, já que as informações trafegavam em alguns casos de forma analógica, geralmente unidirecional, permitindo apenas o tráfego de áudio e vídeo necessários à retransmissão. Nos postos de telefonia havia a possibilidade de dados adicionais que proporcionavam algum tipo de controle. Em ambos os casos, se algum problema ocorresse, não poderia ser percebido rapidamente.

Apesar do avanço da tecnologia, o custo de monitoração de algum posto remoto é elevado. Poucas soluções estão disponíveis. Por outro lado, novos equipamentos estão sendo lançados. São equipamentos microprocessados, de baixo custo que permitem o desenvolvimento de uma grande gama de aplicações. É nesta linha que se pretende trabalhar. Aliar a grande disponibilidade de pequenos módulos de desenvolvimento, software livre e oportunidade de propor uma solução eficiente e barata. Monitorando e gerenciando remotamente um posto de retransmissão, seja ele de televisão ou telefonia, obtendo informações em tempo real com envio de alarmes

em caso de anormalidade. A solução da qual se tratará neste trabalho terá o nome de Telesite.

Esta solução tem, além do que foi citado acima, outras características. A principal delas é a parametrização, ou seja, a capacidade de ser configurada segundo a necessidade do seu usuário e de acordo com limites operacionais de cada sensor aplicado nas suas entradas. Com isso, a solução é capaz interpretar uma gama imensa de sensores e de monitorar os mais diversos equipamentos existentes no mercado.

Em virtude desta capacidade, outros benefícios como rapidez e economia são alcançados, uma vez que será desnecessária a contratação de uma empresa especializada para alterar linhas de comando quando um novo sensor for acoplado ou quando um novo equipamento for implementado num sistema.

Outro fator relevante que faz da solução uma ferramenta útil é a sua aplicabilidade imediata, porque não requer nenhuma adaptação no equipamento a ser monitorado, tornando-o versátil para se adaptar de forma transparente sem ajustes técnicos.

Outra grande vantagem está no fato de o Telesite se basear num produto comercial, o *Raspberry*. Em caso de falhas, este *hardware*, pode ser substituído, sem necessidade de novas configurações, pois todo o sistema operacional e *software* embarcado, bem como os dados lidos anteriormente, ficam armazenados em um cartão de memória.

Com isso, numa visão sistêmica, a empresa ganha em produtividade, economia e eficiência. Isso porque, os técnicos têm um controle maior sobre os postos de retransmissão e podem antecipar necessidade de manutenções. Desta forma o gerenciamento fica menos sobrecarregado, aumentando a eficiência administrativa, diminuindo significativamente deslocamentos para verificações dos postos e conseqüentemente, os custos.

1.1 PROBLEMA

As dificuldades encontradas atuando nas áreas de telecomunicações e televisão são mais percebidas durante as manutenções, devido à falta de informação de como está se comportando algum equipamento ou posto de retransmissão. Situados muitas vezes em locais distantes e de difícil acesso, também com baixa possibilidade de comunicação, fazem com que a solução de uma falha demore um tempo maior do que o previsto. A ausência de monitoramento ocasiona muitas vezes, idas até o local para verificação exata do problema, sem uma estimativa de quantidade de pessoal e quais recursos são suficientes.

Desta forma percebeu-se a necessidade de monitoração e gerência remota destes locais e, devido à tecnologia disponível e em crescente evolução, propõe-se uma solução tecnológica de baixo custo, baseada em uma pequena placa de desenvolvimento com alta capacidade de processamento, com software livre embarcado. A proposta baseia-se no uso de sensores, capazes de obter informações da estação de retransmissão, interligados com a unidade de desenvolvimento, onde as alterações de estado, ou leituras podem ser programadas para condições de falha e estas serem informadas ao setor ou equipe responsável. Através de rede celular, o posto poderá ser acessado remotamente e, recebendo informações do local em tempo real, o gestor responsável pode tomar uma decisão mais assertiva.

1.2 JUSTIFICATIVA

A sugestão proposta facilitará a gerência da estação de forma remota, permitindo saber quando ocorre algum evento anormal. A maior relevância desse trabalho está em garantir uma maior agilidade na percepção e solução de problemas técnicos ocorridos na estação remota, fazendo com que sejam levados materiais específicos para a solução do problema.

Em termos financeiros o projeto é de custo muito baixo, uma vez que se baseia em hardware barato e software livre. Após a implantação numa estação remota, terá apenas o custo da rede de dados.

Para as empresas de TV, telefonia e afins o equipamento pode ser parametrizado para cada caso específico. As portas de detecção são configuráveis,

de acordo com os parâmetros que se deseja monitorar usando para isso sensores adequados.

1.3 OBJETIVOS

1.3.1 Geral

Desenvolver uma solução tecnológica visando monitorar remotamente um posto de retransmissão de Televisão ou Telecomunicações.

1.3.2 Objetivos específicos

Desenvolver uma forma de solucionar um problema de falta de informações acerca de um posto de retransmissão, de TV ou Telecom, localizado em local onde não há gerenciamento remoto.

Monitorar num posto de retransmissão, falhas e alarmes e analisar dados obtidos.

2 METODOLOGIA

Constatadas as dificuldades encontradas na prática para fazer a supervisão de locais onde há um posto de retransmissão, seja pela distância ou pela deficiência de infraestrutura, será buscada uma solução para possibilitar a monitoração remota desses locais. Para saber o que e como monitorar serão usadas fontes de pesquisas bibliográficas e documentais sobre telecomunicações, bem com a experiência técnica dos autores.

Tomando por base a tecnologia hoje existente, especialmente na questão da internet das coisas, será desenvolvido um sistema que fará a leitura de parâmetros por meio de sensores. Com o processamento dos valores obtidos, através de entradas de sensores, será possível configurar valores para disparo de alarmes e envio de alertas para o gestor responsável, em tempo real. Esta solução será de grande importância para monitorar postos, onde hoje não se tem acesso às informações sem que haja deslocamento ao local. Com relação à aplicação prática, será usada a metodologia aplicada com testes com sensores reais e compilação de dados coletados para geração e tratamento de conteúdo de estudo.

Para o desenvolvimento prático foi escolhida a plataforma de hardware *Raspberry Pi*, em sua versão 2. Trata-se de um minicomputador de baixo custo para desenvolvimento de soluções de forma simples. Ele tem capacidade para rodar um sistema operacional e conseqüentemente, aplicativos gerados ou instalados, lendo entradas físicas que fazem parte deste hardware e possibilitam leitura de dados externos, através de sensores que traduzem as características do posto monitorado. Dentre outras opções foi o que melhor atendeu as necessidades do projeto em termos de recursos, custo e bibliografia de referência.

A parte de hardware contará também com uma placa para entrada de dados digitais e outra para entrada de dados analógicos. Nas entradas digitais serão lidos dados de sensores que disponibilizarão informações de estados, como por exemplo, uma porta aberta ou fechada, um equipamento ligado ou desligado. Nas entradas analógicas serão lidos valores de uma grandeza, como uma tensão elétrica, uma temperatura de um ambiente ou potência de um transmissor. Ambas as placas terão

conectores para ligação de sensores. Haverá também uma fonte de alimentação para prover energia para o circuito e também para alguns sensores.

No hardware será instalado o sistema operacional *Raspbian*, uma distribuição *Debian* do Linux para o *Raspberry*. O código para leitura de sensores e todo o processamento inicial de dados será desenvolvido em linguagem de programação *Python*. Será desenvolvida uma interface gráfica em *PHP*, para mostrar os dados e leituras obtidas. A interface gráfica será executada a partir do servidor de aplicação *Apache*, instalado no próprio *Raspberry*, que armazenará ainda um banco de dados implementado em *SQLite*.

O sistema contará com meios de comunicação, acessando a internet, para possibilitar a interligação de um computador ao posto de retransmissão. Esta conexão poderá ser feita com a utilização da rede de dados de um *modem 3G*. Pretende-se implementar uma *VPN* (rede virtual privada) com o computador de monitoração. Os dados serão armazenados no próprio equipamento e acessados quando necessário, no computador de monitoração mediante uso de um navegador de internet.

Poderão ser disponibilizados e armazenados dados em nuvem, para acesso a partir de qualquer local pela internet.

3 FUNDAMENTAÇÃO TEÓRICA

3.1 SISTEMAS DE COMUNICAÇÕES

Comunicação significa a transmissão de informação de um ponto a outro. (HAYKIN, MOHER, 2011)

A comunicação pode ser conceituada como uma rede por meio da qual fluem as informações que permitem o funcionamento da estrutura de forma integrada e eficaz. Para um sistema de telecomunicações, onde também se encaixa o Telesite, é a possibilidade de se conectar um site de retransmissão a um sistema de monitoração. Um local que se deseja monitorar deve possuir os sensores instalados e uma central concentradora dos dados lidos. Entretanto, para que estes dados possam ser obtidos por uma central de monitoração, eles precisam chegar até ela. Nesta etapa é que se faz importante um sistema de comunicação.

A forma mais simples de fazer a conexão da estação de monitoração ao local ou ao equipamento monitorado, seria por um par de fios, mas isso traz algumas desvantagens e até impeditivos. A distância geralmente é elevada e desta forma torna-se inviável tal instalação, seja pelo gasto ou pela perda de qualidade de sinal causada pela distância. Outro fator importante observado nesta instalação, caso ela seja possível, é a proteção elétrica que pode ser uma vulnerabilidade, pois pode ocorrer danos elétricos causados pela diferença de potencial ou por descargas atmosféricas. Uma boa opção para este caso é a utilização de fibras óticas, porém o custo pode inviabilizar seu uso, dependendo da importância das informações. Para isso, existem outras formas de conexão mais eficientes, descritas a seguir.

3.1.1 Meios sem fio

Os meios sem fio podem prover a comunicação de uma unidade monitorada com a central de monitoração de diversas formas e tecnologias. Dentre elas, podem ser citadas, rede de celular, Bluetooth, rádio enlaces ou comunicação via satélite, bem

como tecnologias derivadas destas. A utilização de cada uma destas tecnologias é definida de acordo com a importância da estação monitorada, custo e disponibilidade de dados. Devem ser avaliadas as distâncias envolvidas, o custo, volume de dados, disponibilidade de cada meio na região e qualidade de serviços, para garantir de forma constante o acesso ao meio que se está monitorado.

3.1.2 Central de Monitoração

Central de monitoração é um local onde chegam as informações obtidas de um ou mais equipamentos. Neste local a monitoração pode ser realizada por sinalizações visuais ou sonoras, por um simples computador, que mostra em tempo real as informações de interesse ou por diversos monitores capazes de mostrar informações de vários locais dentro de uma gama de estações monitoradas. Nesta central podem ser avaliados os dados de equipamentos e situações de falhas e a partir destes dados, podem ser tomadas as decisões e definidas ações para sanar as falhas, com maior rapidez, podendo até evitar interrupções no serviço. (MIJATOVIC, Freddy, 2010)

3.1.3 Telemetria

A palavra telemetria se originou a partir do grego *tele*, que significa “remoto”, e *metron*, que quer dizer “medida”. A telemetria é simplesmente a medição e análise de dados e informações à distância, com a transferência de dados coletados para o monitoramento, medição e controle (VISSOTTO, Junior, 2004).

Em outros termos, numa visão bem objetiva, significa obter leitura de parâmetros de um local próximo ou distante, de modo a facilitar a operação de um equipamento ou de uma estação, obtendo informações importantes sobre algo, como por exemplo, condições meteorológicas.

Cada vez mais a telemetria vem se tornando importante. A demanda por ações rápidas e precisas, faz com que ela torne cada vez mais necessária no controle de

máquinas e estações provedoras de serviços. A tomada de decisões baseadas em minimização e redução de falhas, torna necessários controles cada vez mais precisos sobre parâmetros de trabalho de um equipamento ou serviço.

Um exemplo que pode ser citado é a leitura de consumo de energia elétrica em grandes empresas, não sendo necessário que um leiturista vá até o local onde está instalado o medidor. A leitura pode ser obtida de maneira fácil pela administração da concessionária de energia, via rede de dados e com auxílio de um equipamento sensor instalado no medidor de energia.

As informações meteorológicas são outro exemplo de telemetria, onde diversos sensores captam informações do clima e os dados são concentrados numa central para análise e processamento.

Competições de velocidade por veículos automotores também fazem uso da telemetria. Enquanto o veículo é pilotado, a equipe de apoio pode monitorar todo seu rendimento e as condições de funcionamento. Isso implica em uma tomada de decisões com maior probabilidade de acerto, considerada a disponibilidade dos dados do veículo em tempo real.

Uma infinidade de exemplos pode ser citada para descrever a utilidade de um sistema de telemetria, mas de uma forma geral todos partem de um mesmo princípio. O sistema é composto por sensores, concentrador de dados, meios de comunicação e unidade de gerenciamento.

Segundo TEIXEIRA, Oliveira e Heleno (2014), há alguns anos a telemetria referia-se, apenas à operação com telêmetros, instrumentos ópticos para medir distância de interesse de um analisador e transmissão de dados para posterior análise. Com o avanço tecnológico, o termo telemetria passou a ser empregado também como forma de medição a distância em diversas áreas, desde o automobilismo e aviação, passando pela agricultura e medicina até a biologia. A instalação em locais de difícil acesso, possibilita o monitoramento constante destes sistemas em outro local, possuindo diferentes necessidades de distância e banda de transmissão.

3.2 IOT – Internet das Coisas

Ápice da evolução tecnológica, objetos conectados são apenas o primeiro passo para uma solução de Internet das Coisas (IoT). A conexão entre “coisas” permitirá a otimização e automação de processos. As maiores transformações virão com os objetos inteligentes que levarão a um ganho brutal de produtividade na economia. (CLEAM, Wesley).

Nos Estados Unidos, por exemplo, grande parte dos automóveis estão saindo da fábrica conectados a um serviço de monitoramento durante o período de garantia. A grande transformação virá, no entanto, com o carro de direção autônoma.

A Internet das Coisas pode ser vista como uma evolução, para os estágios de otimização e autonomia da comunicação Máquina-a-Máquina, mais focada em monitoração e controle.

À medida que os processos de uma solução forem automatizados e os objetos se tornarem inteligentes se necessitará cada vez menos intervenção humana, o que causará um grande impacto na produtividade. (Porter, 2014).

3.3 RASPBERRY PI

3.3.1 Ambiente de desenvolvimento

O *Raspberry Pi*, figura 1, é um minicomputador do tamanho de um cartão de crédito, criado na Inglaterra, pela *Raspberry Pi Foundation*, com o objetivo de estimular o ensino de informática básica nas escolas e universidades. Trata-se de um equipamento de baixo custo, com um preço de apenas US\$ 35,00 (trinta e cinco dólares) em seu modelo de melhor desempenho, a versão 2B (*Raspberry Pi 2*).

Figura 1 - *Raspberry Pi 2*



Fonte:Raspberrypi.org

O *Raspberry Pi 2* tem processador *quad-core ARM Cortex A7* rodando a 900 MHz, possui 1GB de memória RAM, 1 (um) controlador *Ethernet*, 4 (quatro) portas USB 2.0 e 40 (quarenta) pinos de GPIO (*General Purpose Input/Output*), que são basicamente, pinos programáveis de entrada e saída de dados. Ele conta ainda com processador gráfico, *slot* para cartões de memória, interface USB, HDMI e seus respectivos controladores. Além disso, ele também apresenta entrada de energia e barramentos de expansão. Ainda que minúsculo, o *Raspberry* é um computador completo (Alencar, 2015).

Devido ao tipo de *hardware* de baixo consumo, o *Raspberry* não é o computador ideal para jogos e para *softwares* de edição de vídeo, por exemplo. Ao contrário de um poderoso e enorme *desktop*, ele conta com processadores construídos a partir dos projetos ARM, que são ideais para máquinas genéricas, sistemas de controle e unidades que geram menos calor e gastam menos energia.

Apesar da simplicidade, o *hardware* do *Raspberry Pi* suporta distribuições Linux e pode reproduzir vídeos de alta definição em uma tela de TV via saída HDMI. O usuário pode conectar mouse e teclado comuns para usar o computador. Um exemplo da conexão de periféricos pode ser visto na figura 2.

Figura 2 - Periféricos conectados ao *Raspberry*



Fonte: Tecmundo.

O sistema operacional é instalado em um cartão de memória SD, já que o computador não apresenta disco rígido próprio. Com boa performance, o sistema operacional do *Raspberry Pi* permite navegar na internet, escrever textos, ver vídeos, ouvir música, criar planilhas e realizar praticamente qualquer tarefa possível num computador convencional. Toda a documentação está disponível em código aberto na página da Fundação de mesmo nome. Isso permite que terceiros façam suas versões do *Raspberry Pi*, com modificações que, inclusive, o tornem mais barato e funcional que o original. (LOMAS, Pete, 2013)

3.3.2 Programação do *Raspberry*

A programação inicial exige que sejam conectados alguns periféricos para facilitar a instalação e configuração da placa. Precisa-se basicamente; teclado,

mouse, fonte de alimentação, monitor, cartão micro SD de pelo menos 8 GB e um computador com entrada para cartão SD.

A preparação do cartão SD com o sistema operacional deve ser a primeira etapa a ser feita. No site “www.raspberrypi.org” pode ser obtido o *NOOBS*, um pacote que contém várias versões do SO, com uma interface que facilita o processo de instalação.

Após gravação do *NOOBS* no cartão SD e inserção deste cartão O *Raspberry* pode ser ligado. Ele será inicializado e irá abrir uma interface com o *Raspbian* que foi instalado no cartão. A partir desse ponto a tela principal do sistema operacional sempre poderá abrir direto quando for ligado. (LOMAS, Pete. 2013)

3.3.3 Ferramentas Instaladas

3.3.3.1 Wampserver

WampServer é uma aplicação que instala um ambiente de desenvolvimento *web* no Windows. Com ele é possível criar aplicações *web* com Apache2, PHP e banco de dados *MySQL*. (BRITO, Edivaldo. 2013)

No servidor Apache2 que é um dos aplicativos instalados pelo *Wampserver*, serão executados os softwares cujo desenvolvimento será necessário para a realização do projeto Telesite. Nele também serão salvos os dados de configurações e de leituras de sensores para posterior análise.

3.3.3.2 SQLite

SQLite é uma biblioteca de código aberto (*open source*) desenvolvida em linguagem C, que permite a disponibilização de um pequeno banco de dados na própria aplicação, sem a necessidade de acesso a um SGDB (Sistema de Gerenciamento de Banco de Dados) separado. A estrutura de banco junto com a

aplicação é denominada de “banco de dados embutido” e é indicada para aplicações de pequeno porte que utilizam poucos dados.

A grande vantagem dos bancos de dados embutidos está em sua simplicidade: é mais prático de administrar do que a implementação de SGBD separados, utilizando soluções como SQL Server e Oracle. Por outro lado, o desempenho e a limitação de recursos são desvantagens do *SQLite* e soluções semelhantes. (Fonte: *SQLite*)

3.3.3.3 HTML

Criada por Berners Lee, HTML é uma das linguagens utilizada para desenvolver páginas *web*. O acrônimo HTML vem do inglês e significa *Hypertext Markup Language* ou em português, linguagem de marcação de hipertexto. (EIS, Diego. 2011).

O HTML é uma linguagem baseada em marcação. Todos os elementos que serão exibidos na página são previamente marcados com *tags* e o documento resultante será interpretado pelo navegador que exibirá os dados. O HTML é a linguagem base da internet. Foi criada para ser de fácil entendimento por seres humanos e também por máquinas.

3.3.3.4 Raspbian

O *Raspbian OS* é uma distribuição Linux criada para rodar nos *Raspberry Pi*. Derivada do Debian, essa distribuição é considerada o sistema operacional padrão do computador da *Raspberry Foundation*. Completo e com diversos *softwares* de desenvolvimento, profundo controle sobre o *hardware* da placa, além de ferramentas de acesso à internet, de escritório na forma de *Libre Office* e de entretenimento, o *Raspbian* é o ponto de partida ideal para quem está começando a usar esta placa.

3.3.3.5 Python

A linguagem de programação Python é muito interessante como primeira linguagem de programação devido a sua simplicidade e clareza. Embora simples, é também uma linguagem poderosa, podendo ser usada para administrar sistemas e desenvolver grandes projetos. É uma linguagem clara e objetiva, pois vai direto ao ponto, sem rodeios.

Python é *software* livre, ou seja, pode ser utilizada gratuitamente, graças ao trabalho da Python Foundation e de inúmeros colaboradores. Esta linguagem pode ser utilizada em praticamente qualquer arquitetura de computadores ou sistema operacional.

Python vem crescendo em várias áreas da computação como inteligência artificial, banco de dados, biotecnologia, animação 3D, aplicativos móveis (celulares), jogos e mesmo como plataforma web. (MENEZES, 2010, p. 21).

3.3.3.6 VPN

Uma Rede Virtual Privada (Virtual Private Network - VPN), como o próprio nome sugere, é uma forma de conectar dois computadores utilizando uma rede pública, como a Internet. Como exemplo, uma empresa que precisa interligar duas de suas unidades, deve pensar em algumas alternativas para solucionar o problema:

- 1- Comprar equipamentos sem fio e conectar as unidades por meio de um link de rádio.
- 2- Conectar as duas unidades por meio de um cabo de rede, o que pode ser totalmente inviável dependendo da distância entre estas.
- 3- Pagar uma linha privada (LP) para que as unidades possam se comunicar.
- 4- Interligar suas unidades via satélite.
- 5- Utilizar uma VPN.

Estes são os recursos mais utilizados por empresas, mas alguns deles podem se tornar financeira ou geograficamente inviáveis. A melhor solução, na maioria dos casos, acaba sendo a VPN, pois seu custo é pequeno se comparado às outras opções.

Como a internet é uma rede pública, é preciso criar mecanismos de segurança para que as informações trocadas entre os computadores de uma VPN, não possam ser lidas por outras pessoas. A proteção mais utilizada é a criptografia, pois ela garante que os dados transmitidos por um dos computadores da rede, sejam os mesmos que as demais máquinas irão receber.

Para criar uma rede VPN não é preciso mais do que dois (ou mais) computadores conectados à Internet e um programa de VPN instalado em cada máquina. (MARTINS, Elaine)

3.4 SENSORES

Sensor é o nome dado ao dispositivo que modifica uma característica intrínseca quando ocorre variação física no meio ambiente externo para o qual o dispositivo é sensível. (MEDEIROS, 2012, p.56).

Os sensores são dispositivos que permitem obter informação do meio e interagir com o mesmo. Deste modo, como os seres humanos recorrem ao seu sistema sensorial para esta tarefa, as máquinas e os robôs requerem sensores para interagirem com o meio em que se encontram.

Quando se desenvolvem computadores capazes de responder ordens de voz, por exemplo, instalam-se microfones que são sensores capazes de captar as ondas sonoras e de transformar em impulsos elétricos. Se estes sensores estiverem associados a outros circuitos, a máquina poderá reagir ao estímulo de acordo com as necessidades do usuário. (SILVEIRA, Cristiano Bertulucci)

3.4.1 Sensores analógicos

São sensores que trabalham variando um sinal elétrico em sua saída proporcionalmente ao valor da grandeza que estão lendo. Nesta categoria podem ser citados os seguintes exemplos: temperatura, umidade e incluídos outros como de luminosidade, potência de transmissão de um equipamento, nível de corrente elétrica consumida por um aparelho, tensão elétrica fornecida pela concessionária de energia, vibração, pressão, entre outros.

3.4.2 Sensores digitais

São sensores que informam o estado de um equipamento como uma porta, que pode estar aberta ou fechada, uma lâmpada ou equipamento que pode estar ligado ou desligado, um sensor de presença que detecta ou não a presença de alguém.

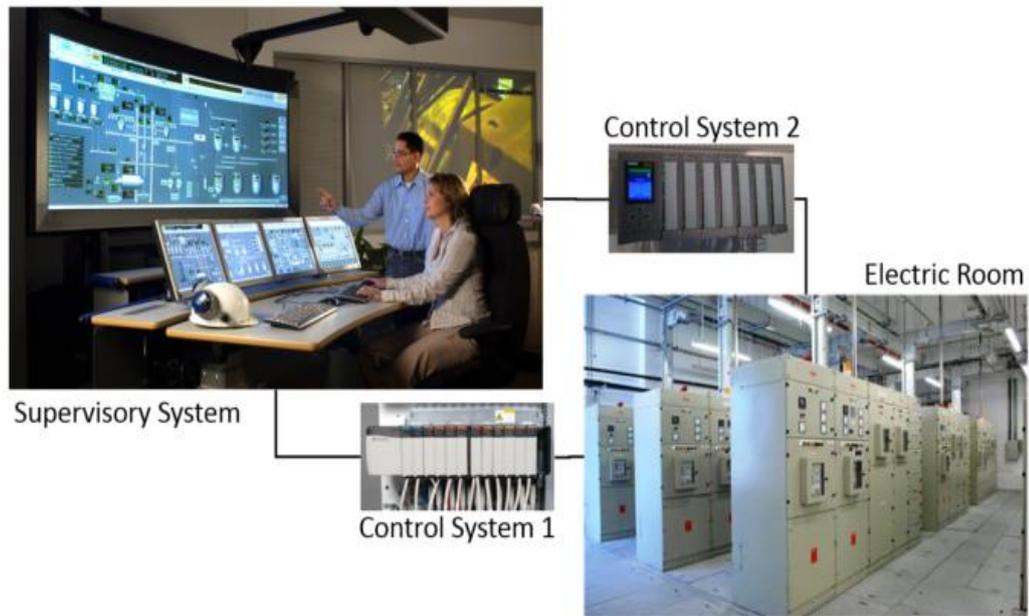
3.5 SISTEMAS DE SUPERVISÃO

Sistemas de controle e supervisão são sistemas independentes e integrados. Os sistemas de controle são responsáveis por controlar máquinas e/ou processos, através de interfaces com os equipamentos de campo. Os sistemas de supervisão são responsáveis por promover a interface entre os sistemas de controle, os usuários e outros sistemas, bem como realizar operações como registro de dados e exibição de alarmes.

Os sistemas de supervisão, também são conhecidos como sistemas SCADA (Supervisory Control and Data Acquisition). Eles são aplicáveis tanto à supervisão de processos industriais quanto à telemetria.

Sendo assim, entende-se que os sistemas de controle podem funcionar independentemente dos sistemas de supervisão, mas os sistemas de supervisão normalmente não podem operar independentemente dos sistemas de controle. Portanto, conforme a figura 3, a estrutura de um sistema supervisório inclui necessariamente os sistemas de controle ou outros sistemas que promovam a interface com os equipamentos físicos. (LEITE, Denis)

Figura 3 - Exemplo de sistema de supervisão.

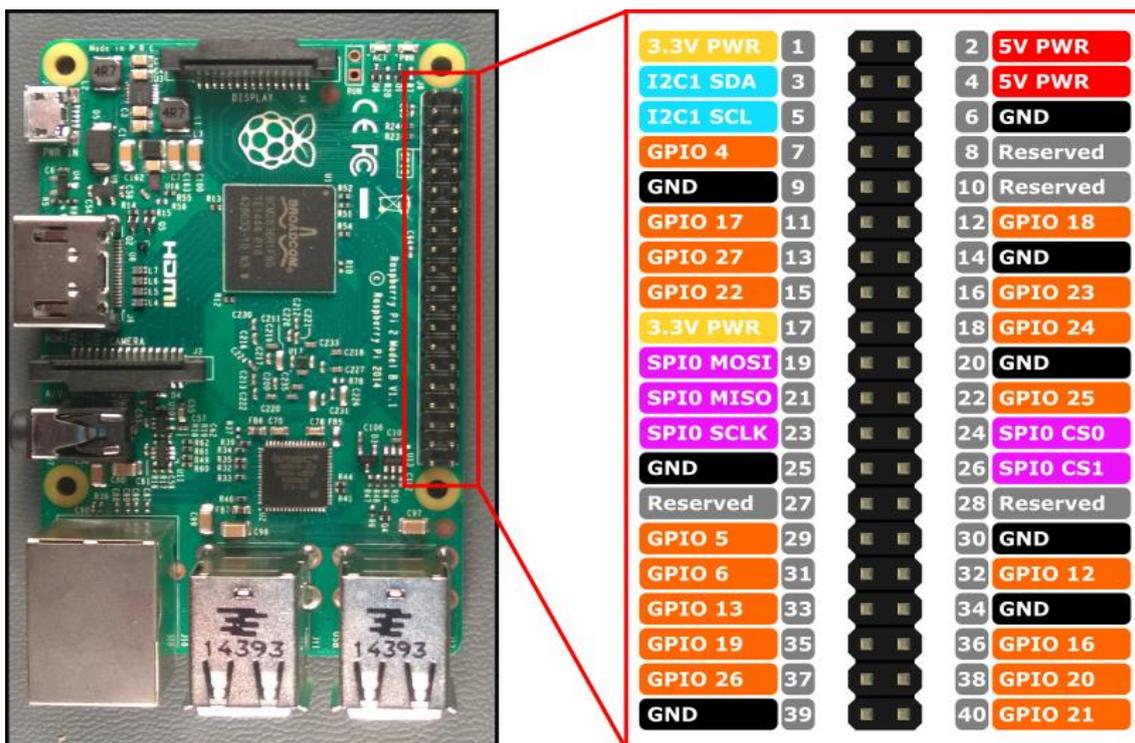


Fonte: mekatronik.

4 DESENVOLVIMENTO

Todo o projeto Telesite foi baseado na utilização do minicomputador *Raspberry*. Este equipamento, de custo muito baixo, face ao que tem a oferecer, tem grande capacidade de processamento e possibilita a conexão de diversos circuitos ao seu barramento. Conhecido como GPIO (*General Purpose Input/Output*), este barramento de uso geral permite conexões de entrada e saída e com isso podem ser conectados diversos sensores, fazendo com que a placa *Raspberry* atenda às necessidades de monitoração para as quais o projeto se propõe. Na figura 4, pode ser vista a placa do *Raspberry* e, em destaque, a descrição das GPIO.

Figura 4 - GPIO (Pinos programáveis de entrada/saída)



Fonte: *Raspberrypi.org*.

Neste *hardware* podem ser instalados sistemas operacionais como o Linux, em várias distribuições. Para o projeto trabalhou-se com o Linux *Raspbian*, versão customizada da distribuição Debian para o *Raspberry*. Para o desenvolvimento da aplicação, foi utilizado o software de desenvolvimento *Python*, que já vem embutido

na instalação padrão do *Raspbian*. A interface gráfica foi implementada em HTML e PHP, rodando em servidor Apache, instalado no equipamento Telesite. Foi utilizado ainda o banco de dados *SQLite*, para armazenamento de dados e configurações de leitura.

Todas estas características possibilitaram o desenvolvimento do projeto com base neste *hardware*.

4.1 FUNCIONAMENTO DO DISPOSITIVO

O monitoramento consiste na leitura de entradas analógicas e digitais através de sensores conectados aos equipamentos e na estrutura do abrigo de Telecom. É possível trabalhar com telecomandos, para fazer acionamento de equipamentos ou comandos remotamente ou mesmo automaticamente de acordo com predefinições ou comportamentos do sistema monitorado.

4.1.1 Leituras analógicas

As leituras analógicas permitem a conexão de sensores cuja saída entrega um nível de tensão que varia proporcionalmente a grandeza medida. O sensor converte um determinado parâmetro lido do equipamento em tensão contínua (DC), proporcional à leitura, variando geralmente entre 0 e 5V. No *Raspberry* não é possível a leitura de valores analógicos. Para isso, utilizou-se um circuito integrado (CI) conversor analógico-digital, o MCP 3008. Este CI pode fazer a conversão de até oito (8) sinais analógicos e entrega a leitura ao Raspberry via protocolo SPI. Cada uma das oito entradas pode ser lida com resolução de 10 bits. Dentre os sensores que podem ser utilizados, pode-se citar: sensor de potência de transmissão, sensor de tensão elétrica, sensor de nível de bateria, sensor de temperatura, de umidade, de corrente elétrica, entre outros.

4.1.2 Leituras digitais

As leituras digitais permitem a conexão de sensores que fornecem estado digital na entrada GPIO da placa, ou seja, pode simbolizar a presença ou a ausência de uma tensão, um estado ativado ou desativado do sensor. A leitura é oferecida por sensor digital que, geralmente possui um contato seco (relé, ou interruptor). Com a alteração do estado de determinado sensor, o *Raspberry* detectará a alteração de algum parâmetro de um equipamento.

Nas leituras digitais podem ser conectados: sensor de abertura de porta, sensor de presença, detector de fumaça, saída de contatos secos de equipamentos, etc.

4.1.3 Telecomandos

O uso de telecomandos possibilita alguma ação no posto, como acionar um determinado equipamento, desarmar um contator (dispositivo eletromecânico que permite o acionamento de cargas que exigem correntes maiores), entre outras finalidades. Basicamente cada comando equivale ao fechamento de uma chave ou interruptor, mas feito através de um relé, podendo ocorrer acionado manualmente pelo operador do sistema através de comando dado pela gerência ou automaticamente vinculado à alguma ocorrência percebida durante a leitura de um sensor.

Esta funcionalidade, embora viável e de implementação razoavelmente simples, está prevista numa atualização futura deste projeto.

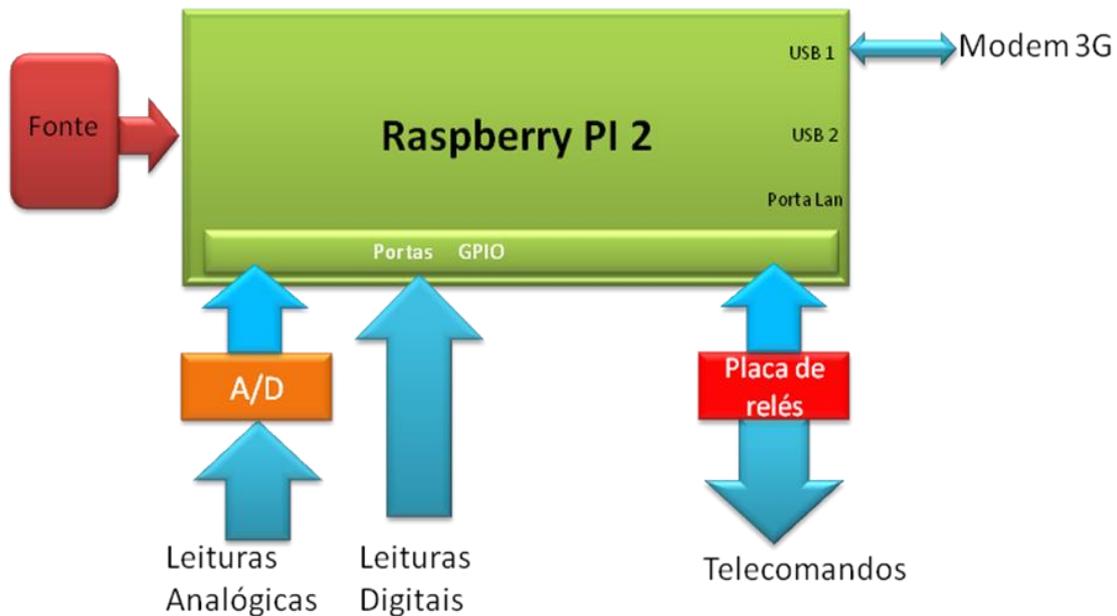
4.1.4 Protocolo SNMP

O protocolo SNMP permite o tráfego de informações de monitoração e controle entre um equipamento e um computador. Neste protocolo, cada parâmetro, possui um identificador, através do qual um valor pode ser solicitado ou alterado pelo sistema de monitoração.

4.2 HARDWARE

O projeto pode ser verificado através do diagrama de blocos da figura 5:

Figura 5 - Diagrama em blocos



Fonte: Imagem Elaborada pelos Autores.

4.2.1 Componentes do *hardware*

- *Raspberry PI 2*;
- Adaptador USB *Wireless*;
- Carregador micro USB;
- Circuito integrado MC3008 (conversor A/D);
- Sensor de tensão (AC/DC);
- Modem 3G;
- *Protoboard* (Matriz de contato);
- Cabo de rede;
- Cabos para *protoboard* (“*jumper*”);

4.2.2 Tratativas durante a elaboração e testes

Para conhecer melhor as necessidades, realizou-se uma visita técnica à Emissora de Televisão RPC (Rede Paranaense de Comunicação), para avaliar a viabilidade técnica e conhecer o que havia de concreto e que pudesse ser usado nas estações remotas para prevenir e informar falhas. Após comprovada a situação real do sistema, concluiu-se que o projeto seria viável e com grande possibilidade de implementação numa unidade de retransmissão de televisão ou telecomunicações, na indústria e até mesmo para uso residencial. Isso foi um fator que aumentou a expectativa inicial da equipe, principalmente pelo fato de se elaborar um projeto, com possibilidades de configuração adaptáveis a diversos tipos de monitoramento.

Definidas as premissas do projeto, começou-se a configuração da placa de desenvolvimento, fazendo a instalação dos *softwares*, configurações iniciais e de acesso remoto ao *Raspberry*. Com base em diversas informações obtidas em materiais apropriados e com o conhecimento técnico da equipe, foi definido um circuito eletrônico capaz de adaptar as entradas de sensores no *Raspberry*, de forma a proteger placa de eventual sobrecarga e também adequar as leituras. Inicialmente, trabalhou-se com montagem em *protoboard*. Foram utilizadas duas placas do *Raspberry* com o intuito de divisão de tarefas do desenvolvimento e mesmo de aprendizado.

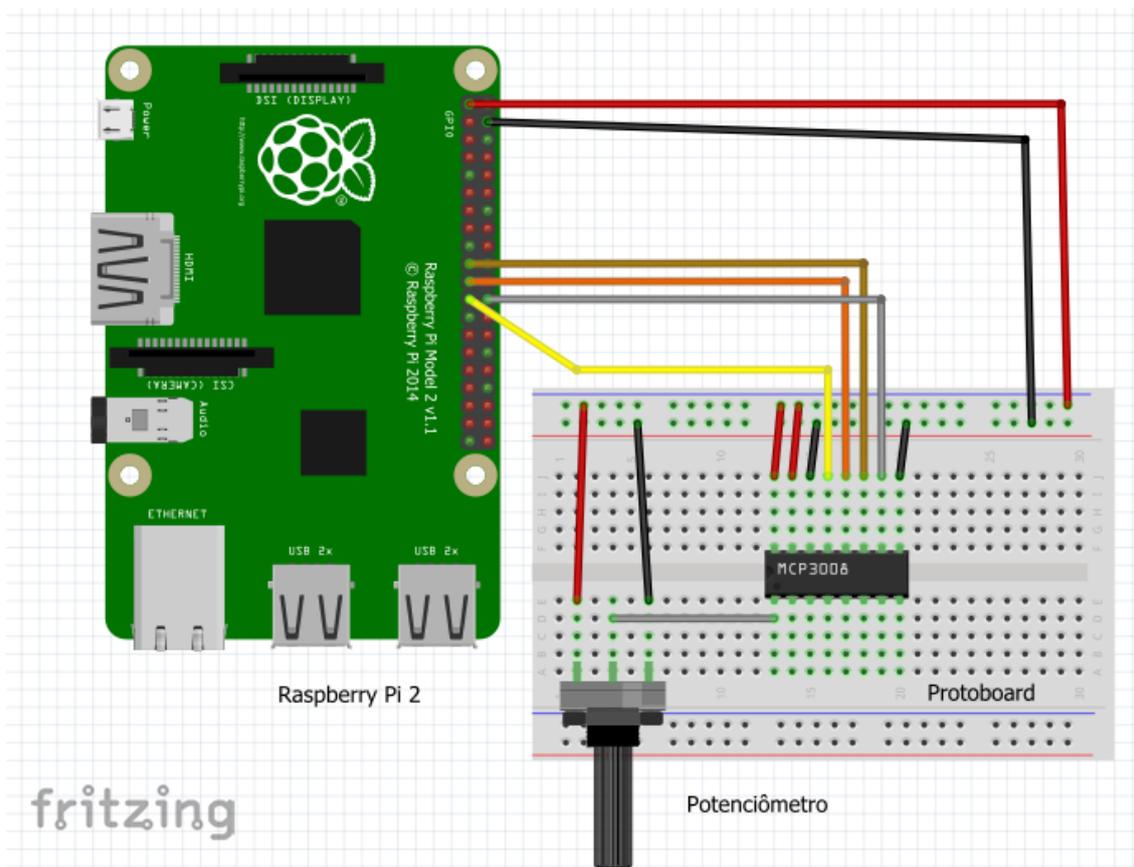
De início os testes foram feitos simulando entradas de sensores. Começou-se o desenvolvimento do programa, implementado em *Python*, para interpretar os valores de tensão nas portas *GPIO* da placa de desenvolvimento. A linguagem utilizada é bem apropriada para esta utilização e, além disso, foi ensinada em disciplina específica do curso.

Para as entradas digitais, ou “*status*”, foi utilizado para cada uma das oito entradas um circuito de *pull-up*, onde um resistor é conectado entre um ponto de tensão do *Raspberry*, 3,3V e a entrada de leitura. Nesta mesma entrada é conectado o sensor que, pode levar a entrada em nível de tensão baixo ou alto através do

fechamento de uma relé, chamado de contato seco. Nos testes simularam-se estes sensores utilizando alguns fios e interruptores para desempenhar a mesma função do contato seco.

Para as entradas analógicas, foi montado um circuito formado basicamente por um circuito conversor Analógico-Digital (A/D), o MCP3008 fabricado pela Microchip. Este circuito possibilitou a utilização de oito canais de entrada, com precisão de conversão de 10 bits. Para a simulação de sensores, utilizou-se um potenciômetro, cujos terminais externos foram interligados à alimentação do circuito e o pino central ao pino de leitura da placa de desenvolvimento, conforme pode ser visto na figura 6.

Figura 6 - Circuito de Leituras Analógicas



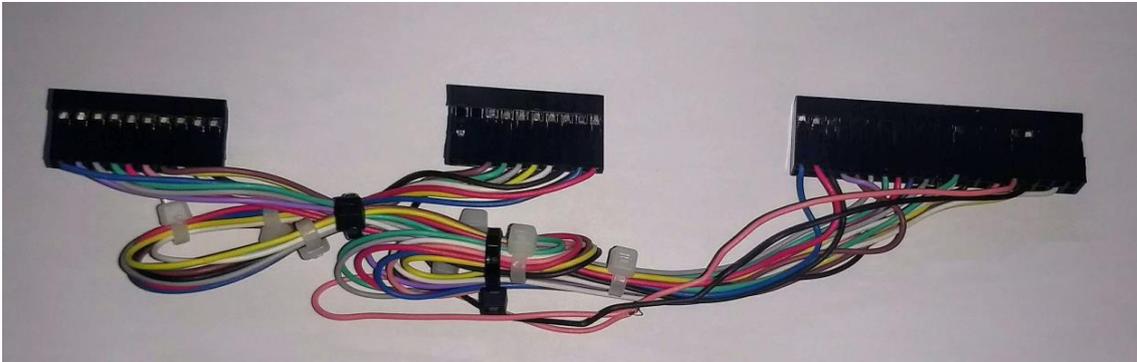
Fonte: Elaborado pelos Autores.

Nesta simulação, a variação da resistência do potenciômetro, altera o valor de tensão lido na porta de entrada analógica, mostrando a atuação de um sensor, que deve fornecer um nível de tensão proporcional à grandeza que ele monitora.

4.2.3 Montagem

Com sucesso nos testes iniciais, começou-se a montagem. Os circuitos foram montados em placa ilhada, com bornes para a conexão de sensores e alimentação dos mesmos. Numa das placas, foi montado o circuito de leituras digitais e no outro o de analógicas. Em um bastidor metálico com tamanho de uma unidade de *rack*, padrão 19 polegadas, foram acomodados o *Raspberry* e as duas placas de entradas de sensores. Foi elaborado um cabo para interligar as placas. Este cabo pode ser visualizado na figura 7.

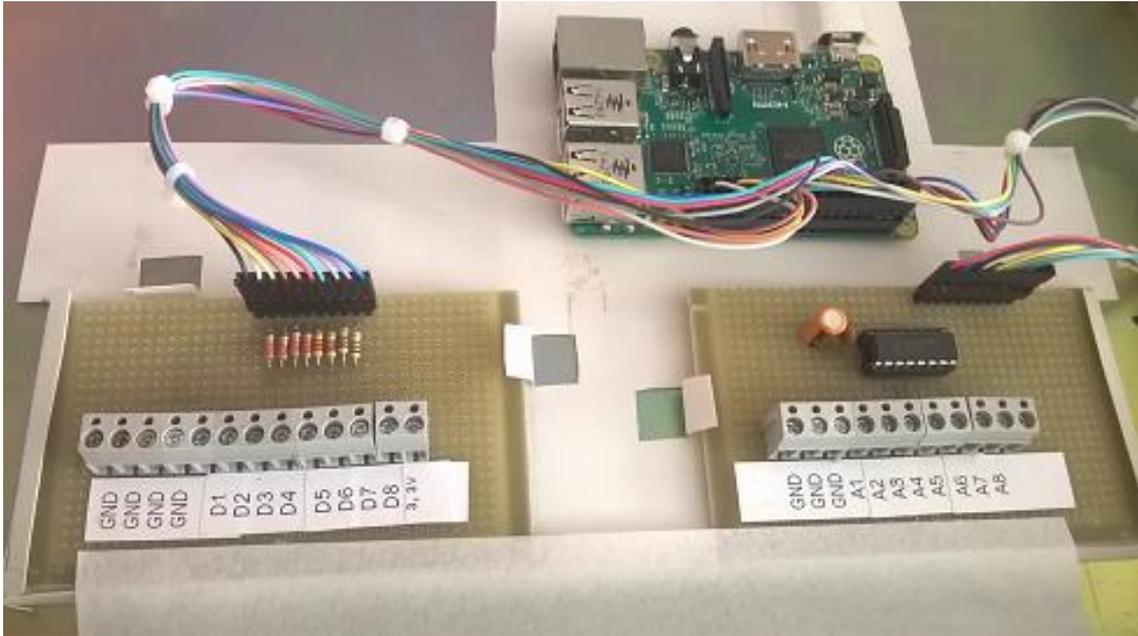
Figura 7 - Cabo de interligação



Fonte: Foto Tirada pelos Autores

A alimentação do sistema foi feita com fonte de 5V (Volts), 2A (Amperes) conectada à rede elétrica. Na figura 8 pode ser visualizado o cabo de interligação instalado nas placas.

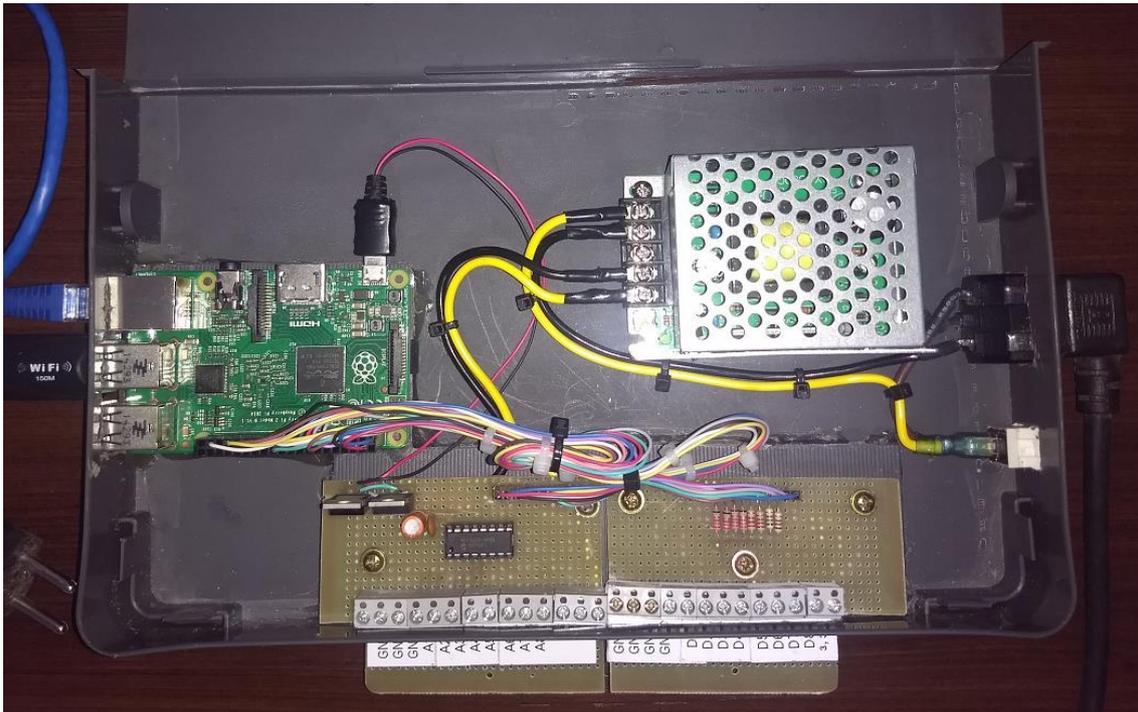
Figura 8 - Placas montadas



Fonte: Foto Tirada pelos Autores

Com o protótipo montado, continuou-se o desenvolvimento do software. Na sequência, montou-se todo o projeto num bastidor plástico, onde ficaram alojadas as placas de entrada de sensores, o *Raspberry* e a fonte de alimentação, todos interligados pelo cabo de interligação. Esta montagem pode ser vista nas figuras 9, 10 e 11.

Figura 9 - Projeto Montado



Fonte: Foto Tirada pelos Autores

Figura 10 - Entrada de Alimentação no Telesite



Fonte: Foto Tirada pelos Autores

Figura 11 - Conexões do *Raspberry*



Fonte: Foto Tirada pelos Autores

4.3 SOFTWARE

Para integrar os sensores com o *Raspberry*, foi utilizado um *software* capaz de operar sobre um *hardware* limitado. Optou-se então pela distribuição de código aberto (*open source*) denominado *Raspbian*, pequeno sistema operacional funcionando sobre uma distribuição OS *Debian* e customizada para equipamentos como o *Raspberry*.

O *software Python*, também foi utilizado na implementação do projeto, pois possui vários recursos para os sistemas embarcados como, por exemplo, interagir com um GPIO. Além disso, permite a criação de vários outros scripts de programação para comunicação através de *e-mail*.

Para acesso remoto ao *Raspberry Pi*, foram utilizados *softwares* com suporte *PuTTY*, um *software* de emulação de terminal grátis e de código livre, o *Filezilla*, para transferência de arquivos e o *Tightvncserver*, que possibilitou acesso remoto à interface gráfica criada posteriormente.

Foi desenvolvido um *software*, em *Python*, para a leitura de todas as entradas de sensores e armazenamento das leituras em banco de dados. Para isso, foi utilizado o banco dados *SQLite*. Ainda no código de leitura, foi desenvolvida a funcionalidade de envio de *e-mail*. Para cada uma das dezesseis entradas de sensor, mediante condição de alarme programável através interface gráfica elaborada em *PHP*, é informada para um e-mail de contato, a ocorrência de alteração das informações dos sensores. Foram testadas condições de alarme, e para todas as entradas de sensores ocorreu o envio de e-mail.

A premissa de funcionamento do sistema é que uma vez ligado o Telesite, o programa obtenha as leituras através de sensores, processe para verificar se estão dentro dos valores configurados, armazene em banco de dados e informe a algum destinatário de e-mail, alguma alteração nas medidas monitoradas.

A configuração das medidas e a visualização das leituras podem ser feita através de página *web*, sobre a qual será tratado na sequência.

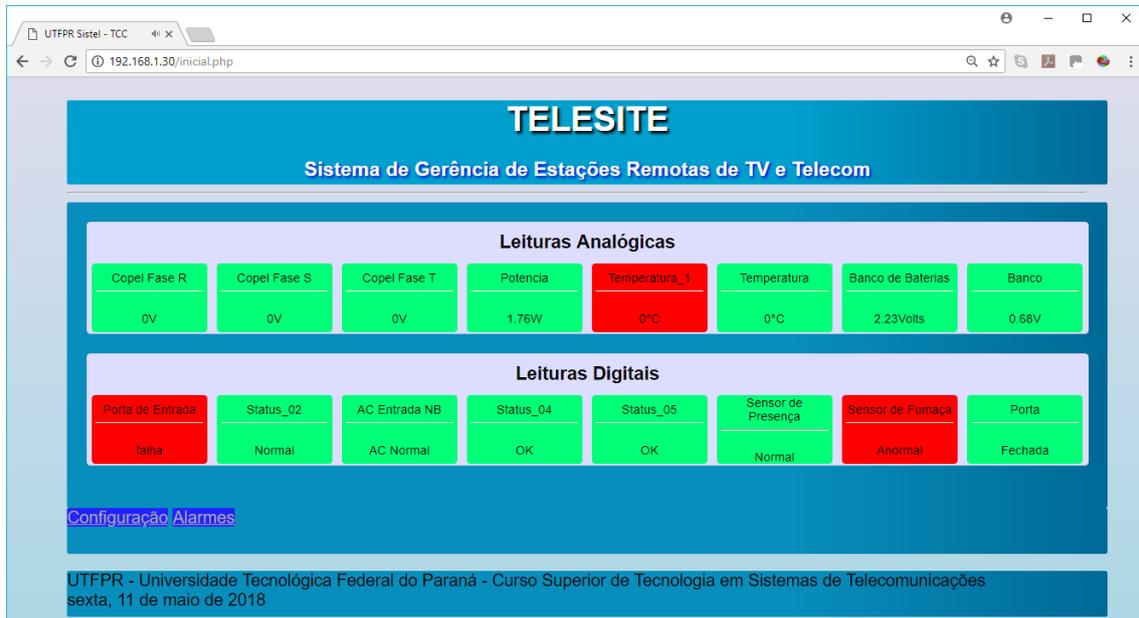
4.3.1 Código desenvolvido para as leituras de sensores

O código desenvolvido para a leitura de sensores foi colocado no apêndice A

4.3.2 Proposta de interface

Para a visualização das leituras de sensores e existência da condição de alarme, foi desenvolvida uma interface em *PHP*. Nesta interface podem ser verificados nomes de cada entrada de sensor e o valor do parâmetro ou situação lida, diferenciando-se a cor para o caso de anormalidade da leitura obtida. Ainda, caso seja acessada esta interface, um aviso sonoro poderá ser ouvido caso esteja ocorrendo um ou mais alarmes. A figura 12, mostra a tela inicial, mas neste caso, sem sensores conectados ao Telesite. Para esta visualização, foram configurados alguns valores para demonstrar algumas leituras em alarme.

Figura 12 - Tela Proposta



Fonte: Cópia de Tela do Sistema.

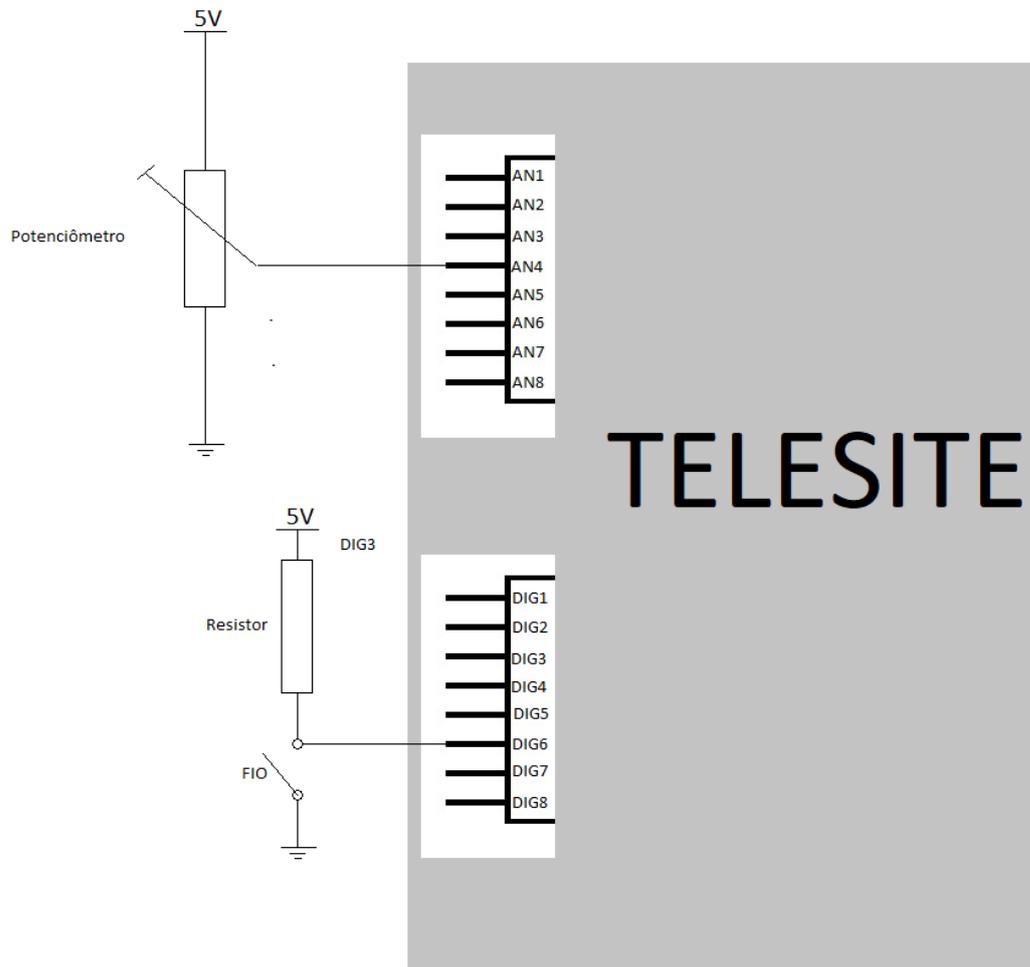
5 APRESENTAÇÃO E ANÁLISE DOS RESULTADOS

Os testes iniciais foram feitos com a montagem do primeiro protótipo, num *proto-board*, composto do *Raspberry* e utilizando um potenciômetro para teste nas leituras analógicas e fios comuns para as digitais. Foram realizados alguns testes com as entradas digitais e analógicas.

Com sucesso nos testes, os circuitos das entradas analógicas e digitais foram montados em placas ilhadas. Também foi montado um cabo para a interconexão destas placas com o *Raspberry*. Desta forma, pôde-se trabalhar no desenvolvimento do software para captação dos dados e armazenamento destes no banco de dados. Também foram implementadas as telas para exibição de dados, alertas de alarmes e configuração das leituras.

Durante o processo de desenvolvimento, trabalhou-se com o potenciômetro para simular as leituras analógicas e um fio para simular as digitais, conforme mostrado na figura 13.

Figura 13 – Simulação de Sensores



Fonte: Imagem elaborada pelos Autores

Com o potenciômetro, foi possível simular o funcionamento de um sensor analógico, através de variação de sua resistência e, conseqüentemente, a tensão nas portas de entrada analógica, entre 0 e 5V, onde o valor de 5V correspondia ao valor máximo da leitura configurada.

Para o circuito das entradas digitais, um fio foi usado para simular o funcionamento de um sensor digital, onde o fio poderia estar conectado ou desconectado da entrada digital.

Através destes testes, reforçou-se a ideia de utilização do Telesite de forma versátil, podendo ser conectados diversos tipos de sensores, que entreguem em sua

saída uma variação de estado, ou de tensão correspondente a grandeza para a qual foram concebidos.

Houve necessidade de diversas alterações nos algoritmos até se chegar à versão final do projeto.

Durante os testes foi notado que o Telesite precisa estar conectado à internet para sincronizar seu horário, o que é necessário para o registro com uma melhor exatidão dos instantes em que ocorrem eventuais registros de alarmes. Como a premissa do projeto é contar com conexão de internet por meio da rede celular, não se trabalhou com a possibilidade de implantar um circuito de relógio em tempo real.

Para configurar uma *VPN* é de fundamental importância ter um *IP* Fixo, foram tentadas alternativas para conseguir atender a esta necessidade. Uma possibilidade de sucesso seria tentar utilizar provedores deste serviço. Tentou-se por algumas vezes fazer as configurações através do provedor *Tunnelbroker.net*, mas o funcionamento esbarrou em uma necessidade de alterar diversas configurações de rede no computador, para possibilitar o funcionamento e com isso, optou-se por priorizar o desenvolvimento da etapa de captura e monitoramento, com conexão direta ao Telesite via cabo de rede, mantendo a necessidade e importância da *VPN*, como implementação futura. Outra possibilidade consiste na contratação de *VPN* pelo cliente que utilize o sistema. Na figura 14 pode ser vista a tela para configuração de *VPN* através do provedor Tunnelbroker.

Figura 14 - Tela do Provedor para VPN

Tunnel Details

IPv6 Tunnel
Example Configurations
Advanced

	Tunnel ID: 475381	Delete Tunnel
	Creation Date:	May 5, 2018
	Description:	VPN TELESITE

IPv6 Tunnel Endpoints

	Server IPv4 Address:	216.66.64.154
	Server IPv6 Address:	2001:470:1f2a:30::1/64
	Client IPv4 Address:	201.24.237.6
	Client IPv6 Address:	2001:470:1f2a:30::2/64

Routed IPv6 Prefixes

	Routed /64:	2001:470:1f2b:30::/64
	Routed /48:	2600:70ff:c807::/48 [X]

DNS Resolvers

	Anycast IPv6 Caching Nameserver:	2001:470:20::2
	Anycast IPv4 Caching Nameserver:	74.82.42.42

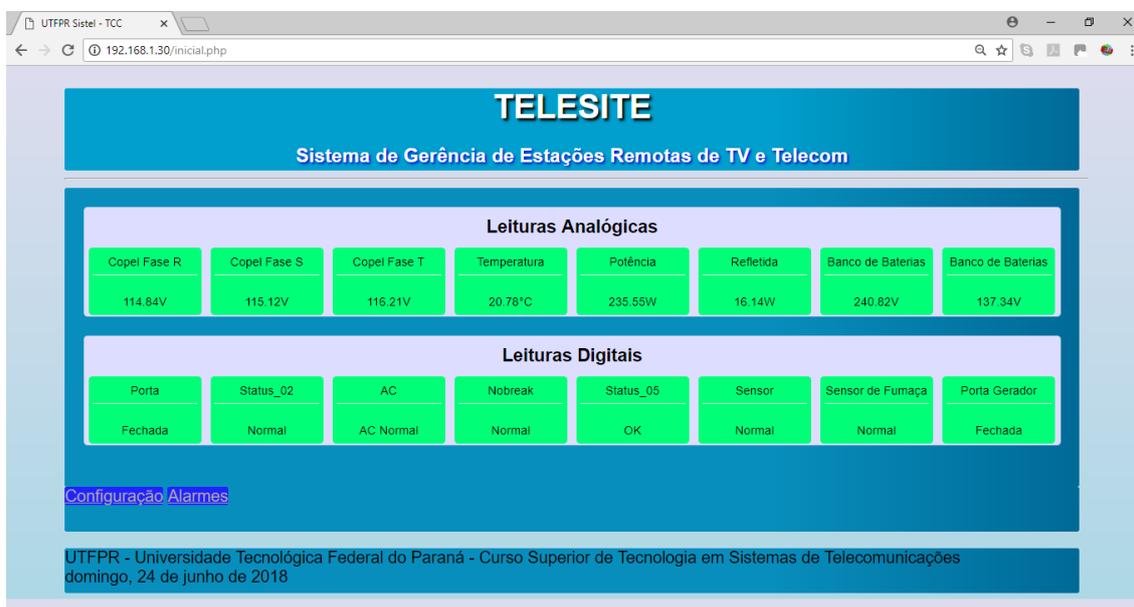
rDNS Delegations [Edit](#)

	rDNS Delegated NS1:	ns1.he.net
	rDNS Delegated NS2:	ns2.he.net
	rDNS Delegated NS3:	ns3.he.net
	rDNS Delegated NS4:	ns4.he.net
	rDNS Delegated NS5:	ns5.he.net

Fonte: Cópia da Tela de Acesso do Provedor do Serviço.

Voltando as preocupações ao projeto em suas premissas básicas, chegou-se a seguinte tela, mostrada na figura 15, acessada no computador de monitoração e hospedada no Telesite.

Figura 15 - Tela Inicial do Telesite



Fonte: Cópia de Tela do Sistema.

Na interface gráfica são mostradas as últimas leituras obtidas dos sensores e salvas no banco de dados. Com o acesso a esta tela, pode-se verificar a atualização das medições conforme elas ocorrem. Em caso de anormalidade em uma ou mais medidas, o respectivo campo, ficará em cor vermelha e será disparado um aviso sonoro até que todas estejam em situação normal.

Todas as leituras analógicas mostradas na tela podem ser configuradas. Através do botão de configuração, pode ser acessada a tela a seguir conforme vemos na figura 16.

Figura 16 - Tela de Configurações do Telesite



Fonte: Cópia de Tela do Sistema

Através desta tela pode-se acessar nova tela para fazer as configurações, bastando escolher a opção que se deseja alterar.

Para as leituras analógicas a tela permite a configuração de nome e unidade da grandeza medida, faixa de medição e valores de alarme, bem como destinatário de e-mail e mensagem para aviso ao gestor do sistema para informar ocorrência de alarmes. Na figura 17 pode ser vista a tela de configuração de entrada analógica.

Figura 17 - Tela de Configuração de Entrada Analógica

The screenshot shows a web browser window with the URL `192.168.1.30/altera.php`. The page title is "TELESITE" and the subtitle is "Sistema de Gerência de Estações Remotas de TV e Telecom". The main content area is a form for editing an analog input. The form fields are as follows:

- Alterar:
- Nome da Leitura: (highlighted in yellow)
- Valor mínimo:
- Valor máximo:
- Unidade:
- Alarme Inferior:
- Alarme Superior:
- E-mail:
- Mensagem de Aviso:

At the bottom of the form, there are two buttons: "Cancelar" and "Salvar". Below the form, a footer bar contains the text: "UTFPR - Universidade Tecnológica Federal do Paraná - Curso Superior de Tecnologia em Sistemas de Telecomunicações".

Fonte: Cópia de Tela do Sistema

Para as leituras digitais, a tela permite a configuração de nome da leitura, nome da situação normal, nome da situação de alarme, valor para detecção de alarme, como destinatário de e-mail e mensagem para aviso ao gestor do sistema para informar ocorrência de alarmes. Na figura 18 pode ser vista a tela de configuração de entradas digitais.

Figura 18 - Tela de Configuração de Entrada Digital

Fonte: Cópia de Tela do Sistema

Ainda na tela de configuração é possível acessar os parâmetros já configurados, tanto para medidas analógicas quanto para digitais, conforme pode ser visualizado na figura 19 e figura 20.

Figura 19 - Tela de Parâmetros Analógicos do Telesite

Medida	Nome	Mínimo	Máximo	Unidade	Alarme Mínimo	Alarme Máximo	e-mail	Mensagem
Analogica_1	Copel Fase R	0	280	V	110	135	claudiopiska@gmail.com	Copel Fase R alterada
Analogica_2	Copel Fase S	1	280	V	110	135	telesiterpi@gmail.com	Copel Fase S alterada
Analogica_3	Copel Fase T	0	280	V	110	135	telesiterpi@gmail.com	Copel Fase T
Analogica_4	Temperatura	0	70	°C	10	50	claudiopiska@gmail.com	Temperatura baixa
Analogica_5	Potência	0	300	W	200	290	telesiterpi@gmail.com	Potência alterada
Analogica_6	Refletida	0	30	W	0	20	telesiterpi@gmail.com	Refletida alta
Analogica_7	Banco de Baterias	0	300	V	200	280	chugan@gmail.com	Banco de Baterias 1 Alterado
Analogica_8	Banco de Baterias	0	160	V	120	150	tekmario@hotmail.com	Banco de Baterias 2 Alterado

Fonte: Cópia de Tela do Sistema

Figura 20 - Tela de Parâmetros Digitais do Telesite

Medida	Nome	Status Normal	Status Alarme	Valor de Alarme	e-mail	Mensagem
Digital_1	Porta	Fechada	Aberta	1	telesiterpi@gmail.com	A porta foi aberta!
Digital_2	Status_02	Normal	Falha	1	telesiterpi@gmail.com	Status 2 em falha
Digital_3	AC	AC Normal	Falta de AC	1	telesiterpi@gmail.com	Entrada de AC ausente
Digital_4	Nobreak	Normal	Bypass	1	chugan@gmail.com	Nobreak Alterado
Digital_5	Status_05	OK	anormal	0	telesiterpi@gmail.com	Status_05 Status_05
Digital_6	Sensor	Normal	Alterado	0	telesiterpi@gmail.com	Sensor 6 alterado
Digital_7	Sensor de Fumaça	Normal	Anormal	0	telesiterpi@gmail.com	Sensor de Fumaça em alarme
Digital_8	Porta Gerador	Fechada	Aberta	0	tekmario@hotmail.com	Porta da Sala do Gerador foi aberta

Fonte: Cópia de Tela do Sistema

Com a implementação das interfaces gráficas, pôde-se realizar um teste usando alguns sensores reais do sistema, com configurações de todos os parâmetros de leitura. Com isso, o programa desenvolvido em *Python*, durante o processo de varredura de sensores, também está apto a analisar as medições e enviar e-mails com informações da leitura em alarme, mesmo que a interface gráfica não esteja sendo acessada.

Para a leitura das entradas analógicas, considerou-se que os sensores deveriam ter ajustes internos do nível de tensão de saída correspondente a grandeza para qual foram concebidos, com valor máximo de 5V. Desta forma, não foi desenvolvida no Telesite, nenhuma etapa de ajuste para compatibilizar a saída do sensor com a entrada de medição. No entanto, foi possível configurar o valor do alcance de medição para que a tensão do sensor pudesse se adequar ao valor de entrada do conversor analógico digital MCP3008, instalado na placa de medidas analógicas. Para estes testes foi utilizado um sensor de tensão alternada, com circuitos internos para detecção de três fases, um sensor de temperatura e um sensor de potência de transmissão, com circuito também para potência refletida.

Para as leituras das entradas digitais não houve problemas, pois, o circuito sensor apenas fecha um contato, informando seu estado de forma semelhante aos testes iniciais, quando se utilizou um fio ligando a entrada de leitura ao ponto de massa

do circuito. Nestas entradas, foi testado sensor de porta, sensor de presença, e duas informações de contato seco, fornecidas por um *no-break*, informando falta de entrada de energia e bateria baixa.

6 CONSIDERAÇÕES FINAIS

Dentre os requisitos propostos pela equipe no início do projeto, conseguiu-se sucesso em praticamente todos os tópicos. Buscou-se neste projeto explorar alguns conceitos bem atuais como internet das coisas, disponibilidade de informações em tempo real, a parametrização para qualquer tipo de sensor ou equipamento, entre outras facilidades necessárias para uma melhor gestão de recursos monitorados. Foi demonstrado que a solução proposta neste projeto pôde ser atendida com baixo custo, eficiência operacional, alta disponibilidade de monitoramento. Ainda assim, são inúmeras as possibilidades de melhorias no *software* do projeto, possibilitando novos recursos de monitoração e administração de dados, buscando atender as necessidades dos gestores da estação monitorada.

Pode ser destacada no Telesite a sua versatilidade para monitorar um sistema qualquer, utilizando para isso os sensores adequados e com parâmetros facilmente configuráveis no sistema. Em caso de falha, o *hardware* já consolidado e de baixo custo, pode ser facilmente substituído, sem necessidade de novas configurações. Outro ponto importante é o baixo consumo de energia, o que permite que o Telesite possa ser alimentado por baterias de pequena capacidade, permitindo a disponibilidade de acesso mesmo sem energia do sistema monitorado.

Pela experiência adquirida neste projeto e pela forma que foi concebido, o mesmo pode ser utilizado na prática. Melhorias podem ser implementadas. Dentre elas, pode-se destacar: otimizar ainda mais os códigos fontes;

- implementar uma *VPN*;
- implementar gerenciamento *SNMP*, como agente e gerente;
- implementar controle através de telecomandos;
- implementar ferramentas para explorar graficamente cada medida em tempo real ou resultados armazenados;
- realizar armazenamento, também em nuvem, dos resultados capturados;
- centralizar, num sistema várias, unidades monitoradas; e

- alimentar o sistema por energia solar.

REFERÊNCIAS

BRITO, Edivaldo. Com WampServer tenha um servidor web completo em seu computador. Disponível em: < <http://www.techtudo.com.br/tudo-sobre/wampserver.html> >. Acesso em: 29 out. 2017.

BUENO, Cleiton. Raspberry PI – GPIO input com Python. Disponível em: <http://www.embarcados.com.br/raspberry-pi-gpio-input-com-python> >. Acesso em: 22 mai. 2017.

CLEAM, Wesley. O que é IoT (Internet das Coisas)? Futuro ou Presente? Disponível em: <https://transformacaodigital.com/o-que-e-iot-internet-das-coisas/> >. Acesso em: 02 mai. 2018.

COMPUTER NETWORKS. Disponível em: < http://www.teraits.com/pitagoras/marcio/gpi/b_ATanenbaum_RedesDeComputadores_4aEd.pdf >. Acesso em: 29 mai. 2017.

Conectando o ADC MCP3008 ao Raspberry Pi para leitura de sensores analógicos, etc. Disponível em: < <http://blog.everpi.net/2014/07/conectando-adc-mcp3008-ao-raspberry-pi.html> >. Acesso em: 10 jun, 2017.

Documentação do Raspberry Pi. Disponível em: <<https://www.raspberrypi.org/documentation> >. Acesso em: 23 fev. 2017.

EIS, Diego. O básico: O que é HTML?. Disponível em: <<https://tableless.com.br/o-que-eh-html-basico> >. Acesso em: 02 mai. 2018.

FELIX, Rafael. SQLite - O pequeno Notável. Disponível em: <<https://www.devmedia.com.br/sqlite-o-pequeno-notavel/7249> >. Acesso em: 02 nov. 2017.

HAYKIN, Simon. MOHER, Michael. Sistemas de Comunicação. São Paulo. Ed. Bookman, 2011.

LEITE, Dênis. Sistemas de controle e supervisão: o que são e como funcionam. Disponível em:< <http://www.mekatronik.com.br/blog/2016/08/28/sistemas-supervisorios-o-que-sao-e-como-funcionam> >. Acesso em: 15 mar. 2018.

MARTINS, Elaine. O que é VPN?. Disponível em: < <https://tecmundo.com.br/1427-o-que-e-vpn-.htm> >. Acesso em: 20 mar. 2018.

MEDEIROS, Júlio C.O. **Princípios de Telecomunicações: Teoria e Prática.** São Paulo: Editora Érica, 2012.

MENEZES, Nilo N.C. **Introdução à Programação com Python**: Algoritmos e lógica de programação para iniciantes. São Paulo: Editora Novatec, 2010.

O que é SQLite?. Disponível em: <<https://www.portalgsti.com.br/sqlite/sobre> >. Acesso em: 04 dez. 2017.

PILGRIM, Mark. **HTML 5**: Entendendo e Executando. Rio de Janeiro: Editora Alta Books, 2011.

PINTO, Pedro. Tutorial Raspberry Pi – Enviar e-mails via Gmail. Disponível em: <<http://pplware.sapo.pt/informacao/tutorial-raspberry-pi-enviar-e-mails-via-gmail> >. Acesso em: 28 mai. 2017.

PYTHON (O que é? Por que usar?). Disponível em: <<http://pyscience-brasil.wikidot.com/python:python-oq-e-pq> >. Acesso em 01 jun. 2017.

RASPBERRY-PI. Disponível em: <<https://www.hardware.com.br/noticias/2015-02/raspberry-pi.html> >. Acesso em: 29 mai. 2017.

RASPBERRY-PI. Disponível em: <<https://www.tecmundo.com.br/campus-party-brasil-2013/36234-criador-do-raspberry-pi-palestra-na-campus-party-2013.html> >. Acesso em: 29 mai. 2017.

SILVA, Regis da. Gerenciando banco de dados SQLite3 com Python - Parte 1. Disponível em: <<http://pythonclub.com.br/gerenciando-banco-dados-sqlite3-python-parte1.html> >. Acesso em: 08 nov. 2017.

SILVA, Regis da. Guia rápido de comandos SQLite3. Disponível em: <<http://pythonclub.com.br/guia-rapido-comandos-sqlite3.html> >. Acesso em: 02 nov. 2017.

SILVEIRA, Cristiano Bertulucci. Sensor: Você sabe o que é quais os tipos. Disponível em: <<https://www.citissystems.com.br/sensor-voce-sabe-que-quais-tipos> >. Acesso em: 01 jun. 2017.

SQLite Tutorial. Disponível em: <<http://www.sqlitetutorial.net/sqlite-php/query> >. Acesso em: 08 nov. 2017.

TAVARES. Inicializando um programa automaticamente no RaspberryPI. Disponível em: <<http://cadernodelaboratorio.com.br/2015/06/10/inicializando-um-programa-automaticamente-no-raspberrypi/> >. Acesso em: 09 ago. 2017.

TEIXEIRA, F.; OLIVEIRA, M. C.; HELLENO, A. L. Telemetria Automotiva via Internet Móvel. Revista Unisal, Americana, v. 16, n. 28-29, p. 1- 10, 2014.

Tunnel Broker via Hurricane Electric. Disponível em < <http://ipv6.br/post/tunnel-broker-via-hurricane-electric/> >. Acesso em: 22 fev. 2018.

VISSOTTO JUNIOR, D. Transmissão de dados via telemetria: uma opção de comunicação remota. Disponível em: < <https://www.vivaolinux.com.br/artigo/Transmissao-de-dados-via-telemetria-uma-opcao-de-comunicacao-remota> >. Acesso em: 5 mai. 2018.

VNC (VIRTUAL NETWORK COMPUTING). Disponível em: < <https://www.raspberrypi.org/documentation/remote-access/vnc> >. Acesso em: 27 jul. 2017.

VPN (O que é VPN), Disponível em: < <https://www.tecmundo.com.br/1427-o-que-e-vpn-.html> >. Acesso: em 01 jun. 2017.

APÊNDICE A - Código Fonte da Aplicação em Python

```

import RPi.GPIO as GPIO
import threading
import datetime
import time
import smtplib
import spidev
import sqlite3

#_____ Função para envio de e-mail_____
GMAIL_USER = 'telesiterpi@gmail.com'
GMAIL_PASS = 'integrador2016'
SMTP_SERVER = 'smtp.gmail.com'
SMTP_PORT = 587

def send_email(recipient, subject, text):
    smtpserver = smtplib.SMTP(SMTP_SERVER, SMTP_PORT)
    smtpserver.ehlo()
    smtpserver.starttls()
    smtpserver.ehlo
    smtpserver.login(GMAIL_USER, GMAIL_PASS)
    header = 'To:' + recipient + '\n' + 'From: ' + GMAIL_USER
    header = header + '\n' + 'Subject:' + subject + '\n'
    msg = header + '\n' + text + ' \n\n'
    smtpserver.sendmail(GMAIL_USER, recipient, msg)
    print(subject + "----- email enviado!")
    smtpserver.close()
    return

#_____ Ligações Conversor A/D_____
#MCP3008 pino 16 VDD >> Raspberry Pi pino 4 5V
#MCP3008 pino 15 VREF >> Raspberry Pi pino 1 3.3V
#MCP3008 pino 14 AGND >> Raspberry Pi pino 6 GND
#MCP3008 pino 09 DGND >> Raspberry Pi pino 9 GND
#MCP3008 pino 13 CLK >> Raspberry Pi pino 23
#MCP3008 pino 12 OUT >> Raspberry Pi pino 21
#MCP3008 pino 11 DIN >> Raspberry Pi pino 19
#MCP3008 pino 10 CS/SHDN >> Raspberry Pi pino 24

#_____ Função de Leitura de Entradas Digitais_____
def leentrada(a):

    GPIO.setmode(GPIO.BCM)
    GPIO.setup(a, GPIO.IN, pull_up_down=GPIO.PUD_UP)
    while True:
        input_state = GPIO.input(a)
        if input_state == False:
            return 1
        else:
            return 0

#_____ Função de Leitura de Entradas Analógicas_____
spi = spidev.SpiDev()

```

```

spi.open(0,0)
def analogica(an):
    r = spi.xfer2([1, (8 + an)<< 4, 0])
    adc_out = ((r[1]&3) << 8)+r[2]
    return adc_out

#_____Leituras de entradas_____

varredura=10;#contador para limitação de quantidade de registros no
banco de dados
if(leentrada(12)==0):#inicialização do programa(usado apenas no
desenvolvimento.)
    inicia=1;
while (inicia):
    print("Verifica = 1,em funcionamento")

#_____Preparação de uma lista para armazenar registros no banco
de dados

    hoje=datetime.date.today()
    agora=time.ctime()
    tstamp = datetime.datetime.now().timestamp()
    lista_leituras = []#lista para armazenar dados para depois
inserir ao banco
    lista_leituras.append(hoje)#lista - adiciona primeiro valor
    lista_leituras.append(agora)#lista - adiciona segundo valor
    lista_leituras.append(tstamp)#lista - adiciona terceiro valor

#_____busca no banco de dados valores mínimos e máximos
configurados, para medir a entrada analógica dentro dos valores
adequados

    conn = sqlite3.connect('banco.db')
    cursor = conn.cursor()
    cursor.execute("""SELECT nome, minimo, maximo, email, mensagem
FROM parametros""");
    conn.commit();
    an=0;
    for linha in cursor.fetchall():
        nome = (linha[0]);
        minn = (linha[1]);#Recebe valor inferior do Range de
medidas para fazer o calculo da medição#
        maxx = (linha[2]);#Recebe valor superior do Range de
medidas para fazer o calculo da medição#
        dest = (linha[3]);
        horario = 1;
        msg = (linha[4]);

        leitura = analogica(an);
        l = leitura * maxx / 1024;
        assunto = ("Medida - " + nome + " - Em alarme!");

        lista_leituras.append(round (l,2)); #lista - adiciona
valores analógicos
        if (l < (float(minn)))or(l > (float(maxx))):

```

```

        leituranormal = ("Medida - " + nome + " - Em
alarme");
        varredura = 0;
        t = threading.Thread(target = send_email,
args=(dest, assunto, msg))
        t.start()
    else:
        leituranormal = ("Medida - " + nome + " - Situação
normal");
        print(leituranormal);

        an=an+1;

    lista = [26, 13, 6, 5, 22, 27, 17, 4]
    ind=0
    aux=1
    conn = sqlite3.connect('banco.db')
    cursor = conn.cursor()
    cursor.execute("""SELECT nome, alarmeVal, email, mensagem FROM
parametrosdig""");
    conn.commit();
    for linha in cursor.fetchall():
        nome = (linha[0]);
        alarmeVal = (linha[1]);
        dest =(linha[2]);
        msg =(linha[3])

        dig=lista[ind]
        leitdigi=leentrada(dig)
        print ("Digital_ %d Status = %d\t" %( aux, leitdigi))
        aux+=1
        ind+=1

        lista_leituras.append(leitdigi);#lista - adiciona valores
digitais
        if (leitdigi != alarmeVal):
            varredura = 0;
            t = threading.Thread(target = send_email, args=(dest,
assunto, msg))
            t.start()

        hoje=datetime.date.today()
        agora=time.localtime()
        tstamp = datetime.datetime.now().timestamp()
        if (varredura == 0):#se ocorreu alarme em alguma das medidas, ou
ocorreram 10 leituras, o registro de todas armazenado
            conn = sqlite3.connect('banco.db')
            cursor = conn.cursor()
            cursor.execute (""INSERT INTO leituras(data, hora,
datahora,
a1, a2, a3, a4, a5, a6, a7, a8,
d1, d2, d3, d4, d5, d6, d7, d8)VALUES
( ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?)""",
lista_leituras)
            conn.commit()
            conn.close();

```

```
    varredura=10;
    print(varredura);
    print("banco de dados atualizado!")
    print("recomeça a varredura");
else:
    lista_leituras.clear();#apaga lista de leituras
    varredura-=1;

time.sleep(2);
```

APÊNDICE B - Código Fonte da Página Inicial.php

```

<!DOCTYPE html>
<html lang = "pt-BR">
<head>
    <title>UTFPR Sistel - TCC</title>
    <meta http-equiv="Content-Type" content="html/text";
charset="utf-8"/>
    <meta http-equiv='refresh' content='05' />
    <link rel="stylesheet" href="style.css" media="all" />

    <script
src="//ajax.googleapis.com/ajax/libs/jquery/1.11.1/jquery.min.js"></
script>

</head>
<body>

<div class="superior">
    <!---->
    <h1 align=center>TELESITE</h1><h3 align=center> Sistema de
Gerência de Estações Remotas de TV e Telecom</h3>
</div>
<hr>
<!--audio alarme-->

<?php
$som=0;
$sonoro = ("<audio autoplay><source src=beep-04.mp3
type=audio/mpeg></audio>");
?>

<?php //função email
$to      = 'claudiopiska@gmail.com';
$subject = '[Telesite] - Falha Detectada';
$message = 'teste teste teste teste';
$headers = 'From: telesiterpi@gmail.com' . "\r\n" .
    'Reply-To: webmaster@example.com' . "\r\n" .
    'X-Mailer: PHP/' . phpversion();

mail($to, $subject, $message, $headers);
?>

<div class="central"><br>
<?php
include('conndb.php');

$sultimo = "SELECT MAX(ID) FROM leituras";
$querySelect = "SELECT * FROM leituras WHERE id=($ultimo)";
$res=$db->query($querySelect);

while($row = $res->fetchArray()){

    $qSelect = "SELECT * FROM parametros WHERE
medida=('Analógica_1')";

```

```

$result=$db->query($qSelect);

while($row1 = $result->fetchArray()){
    if (($row['a1'] > $row1['almMin']) AND ($row['a1'] <
$row1['almMax'])){
        $an1 = ("<td align=center><button
class=btn_normal>".$row1['nome']."<hr><br>".
$row['a1'].$row1['unidade']."</button></td>");
        $som = ($som OR 0);
    }
    else{
        $an1 = ("<td align=center><button
class=btn_alarm>".$row1['nome']."<hr><br>".$row['a1'].$row1['unidade
']"."</button></td>");
        $som = ($som OR 1);
    }
}

$qSelect = "SELECT * FROM parametros WHERE
medida=('Analógica_2')";
$result=$db->query($qSelect);

while($row1 = $result->fetchArray()){
    if (($row['a2'] > $row1['almMin']) AND ($row['a2'] <
$row1['almMax'])){
        $an2 = ("<td align=center><button
class=btn_normal>".$row1['nome']."<hr><br>".$row['a2'].$row1['unidade
e']"."</button></td>");
        $som = ($som OR 0);
    }
    else{
        $an2 = ("<td align=center><button
class=btn_alarm>".$row1['nome']."<hr><br>".$row['a2'].$row1['unidade
']"."</button></td>");
        $som = ($som OR 1);
    }
}

$qSelect = "SELECT * FROM parametros WHERE
medida=('Analógica_3')";
$result=$db->query($qSelect);

while($row1 = $result->fetchArray()){
    if (($row['a3'] > $row1['almMin']) AND ($row['a3'] <
$row1['almMax'])){
        $an3 = ("<td align=center><button
class=btn_normal>".$row1['nome']."<hr><br>".$row['a3'].$row1['unidade
e']"."</button></td>");
        $som = ($som OR 0);
    }
    else{
        $an3 = ("<td align=center><button class=btn_alarm>

".$row1['nome']."<hr><br>".$row['a3'].$row1['unidade']"."</butto
n></td>");
        $som = ($som OR 1);
    }
}

```

```

    }
}

$qSelect = "SELECT * FROM parametros WHERE
medida=('Analógica_4')";
$result=$db->query($qSelect);

while($row1 = $result->fetchArray()){
    if (($row['a4'] > $row1['almMin']) AND ($row['a4'] <
$row1['almMax'])){
        $an4 = ("<td align=center><button
class=btn_normal>".$row1['nome']."<hr><br>".$row['a4'].$row1['unidade
e']."</button></td>");
        $som = ($som OR 0);
    }
    else{
        $an4 = ("<td align=center><button
class=btn_alarm>".$row1['nome']."<hr><br>".$row['a4'].$row1['unidade
e']."</button></td>");
        $som = ($som OR 1);
    }
}

$qSelect = "SELECT * FROM parametros WHERE
medida=('Analógica_5')";
$result=$db->query($qSelect);

while($row1 = $result->fetchArray()){
    if (($row['a5'] > $row1['almMin']) AND ($row['a5'] <
$row1['almMax'])){
        $an5 = ("<td align=center><button
class=btn_normal>".$row1['nome']."<hr><br>".$row['a5'].$row1['unidade
e']."</button></td>");
        $som = ($som OR 0);
    }
    else{
        $an5 = ("<td align=center><button
class=btn_alarm>".$row1['nome']."<hr><br>".$row['a5'].$row1['unidade
e']."</button></td>");
        $som = ($som OR 1);
    }
}

$qSelect = "SELECT * FROM parametros WHERE
medida=('Analógica_6')";
$result=$db->query($qSelect);

while($row1 = $result->fetchArray()){
    if (($row['a6'] > $row1['almMin']) AND ($row['a6'] <
$row1['almMax'])){
        $an6 = ("<td align=center><button
class=btn_normal>".$row1['nome']."<hr><br>".$row['a6'].$row1['unidade
e']."</button></td>");
        $som = ($som OR 0);
    }
    else{

```

```

        $an6 = ("<td align=center><button
class=btn_alarm>".$row1['nome']."<hr><br>".$row['a6'].$row1['unidade
']."</button></td>");
        $som = ($som OR 1);
    }
}

    $qSelect = "SELECT * FROM parametros WHERE
medida=('Analógica_7')";
    $result=$db->query($qSelect);

    while($row1 = $result->fetchArray()){
        if (($row['a7'] > $row1['almMin']) AND ($row['a7'] <
$row1['almMax'])){
            $an7 = ("<td align=center><button
class=btn_normal>".$row1['nome']."<hr><br>".$row['a7'].$row1['unidade
e']."</button></td>");
            $som = ($som OR 0);
        }
        else{
            $an7 = ("<td align=center><button
class=btn_alarm>".$row1['nome']."<hr><br>".$row['a7'].$row1['unidade
']."</button></td>");
            $som = ($som OR 1);
        }
    }

    $qSelect = "SELECT * FROM parametros WHERE
medida=('Analógica_8')";
    $result=$db->query($qSelect);

    while($row1 = $result->fetchArray()){
        if (($row['a8'] > $row1['almMin']) AND ($row['a8'] <
$row1['almMax'])){
            $an8 = ("<td align=center><button
class=btn_normal>".$row1['nome']."<hr><br>".$row['a8'].$row1['unidade
e']."</button></td>");
            $som = ($som OR 0);
        }
        else{
            $an8 = ("<td align=center><button
class=btn_alarm>".$row1['nome']."<hr><br>".$row['a8'].$row1['unidade
']."</button></td>");
            $som = ($som OR 1);
        }
    }

    $qSelect = "SELECT * FROM parametrosdig WHERE
medidadig=('Digital_1')";
    $result=$db->query($qSelect);

    while($row2 = $result->fetchArray()){
        if ($row['d1'] != $row2['alarmeVal']) {
            $dgl = ("<td align=center><button
class=btn_alarm>".$row2['nome']."<hr><br>".$row2['statusAlarme']."</
button></td>");

```

```

        $som = ($som OR 0);
    }
    else{
        $dg1 = ("<td align=center><button
class=btn_normal>".$row2['nome']."<hr><br>".$row2['statusNormal']."<
/button></td>");
        $som = ($som OR 1);
    }
}

$qSelect = "SELECT * FROM parametrosdig WHERE
medidadig=('Digital_2')";
$result=$db->query($qSelect);

while($row2 = $result->fetchArray()){
    if ($row['d2'] != $row2['alarmeVal']) {
        $dg2 = ("<td align=center><button
class=btn_alarm>".$row2['nome']."<hr><br>".$row2['statusAlarme']."</
button></td>");
        $som = ($som OR 0);
    }
    else{
        $dg2 = ("<td align=center><button
class=btn_normal>".$row2['nome']."<hr><br>".$row2['statusNormal']."<
/button></td>");
        $som = ($som OR 1);
    }
}

$qSelect = "SELECT * FROM parametrosdig WHERE
medidadig=('Digital_3')";
$result=$db->query($qSelect);

while($row2 = $result->fetchArray()){
    if ($row['d3'] != $row2['alarmeVal']) {
        $dg3 = ("<td align=center><button
class=btn_alarm>".$row2['nome']."<hr><br>".$row2['statusAlarme']."</
button></td>");
        $som = ($som OR 0);
    }
    else{
        $dg3 = ("<td align=center><button
class=btn_normal>".$row2['nome']."<hr><br>".$row2['statusNormal']."<
/button></td>");
        $som = ($som OR 1);
    }
}

$qSelect = "SELECT * FROM parametrosdig WHERE
medidadig=('Digital_4')";
$result=$db->query($qSelect);

while($row2 = $result->fetchArray()){
    if ($row['d4'] != $row2['alarmeVal']) {

```

```

                $dg4 = ("<td align=center><button
class=btn_alarm>".$row2['nome'].<hr><br>".$row2['statusAlarme'].</
button></td>");
                $som = ($som OR 0);
            }
            else{
                $dg4 = ("<td align=center><button
class=btn_normal>".$row2['nome'].<hr><br>".$row2['statusNormal'].</
button></td>");
                $som = ($som OR 1);
            }
        }

        $qSelect = "SELECT * FROM parametrosdig WHERE
medidadig=('Digital_5')";
        $result=$db->query($qSelect);

        while($row2 = $result->fetchArray()){
            if ($row['d5'] != $row2['alarmeVal']) {
                $dg5 = ("<td align=center><button
class=btn_alarm>".$row2['nome'].<hr><br>".$row2['statusAlarme'].</
button></td>");
                $som = ($som OR 0);
            }
            else{
                $dg5 = ("<td align=center><button
class=btn_normal>".$row2['nome'].<hr><br>".$row2['statusNormal'].</
button></td>");
                $som = ($som OR 1);
            }
        }

        $qSelect = "SELECT * FROM parametrosdig WHERE
medidadig=('Digital_6')";
        $result=$db->query($qSelect);

        while($row2 = $result->fetchArray()){
            if ($row['d6'] != $row2['alarmeVal']) {
                $dg6 = ("<td align=center><button
class=btn_alarm>".$row2['nome'].<hr><br>".$row2['statusAlarme'].</
button></td>");
                $som = ($som OR 0);
            }
            else{
                $dg6 = ("<td align=center><button
class=btn_normal>".$row2['nome'].<hr><br>".$row2['statusNormal'].</
button></td>");
                $som = ($som OR 1);
            }
        }

        $qSelect = "SELECT * FROM parametrosdig WHERE
medidadig=('Digital_7')";
        $result=$db->query($qSelect);

        while($row2 = $result->fetchArray()){

```

```

        if ($row['d7'] != $row2['alarmeVal']) {
            $dg7 = ("<td align=center><button
class=btn_alarm>".$row2['nome']."<hr><br>".$row2['statusAlarme']."</
button></td>");
            $som = ($som OR 0);
        }
        else{
            $dg7 = ("<td align=center><button
class=btn_normal>".$row2['nome']."<hr><br>".$row2['statusNormal']."<
/button></td>");
            $som = ($som OR 1);
        }
    }

    $qSelect = "SELECT * FROM parametrosdig WHERE
medidadig=('Digital_8')";
    $result=$db->query($qSelect);

    while($row2 = $result->fetchArray()){
        if ($row['d8'] != $row2['alarmeVal']) {
            $dg8 = ("<td align=center><button
class=btn_alarm>".$row2['nome']."<hr><br>".$row2['statusAlarme']."</
button></td>");
            $som = ($som OR 0);
        }
        else{
            $dg8 = ("<td align=center><button
class=btn_normal>".$row2['nome']."<hr><br>".$row2['statusNormal']."<
/button></td>");
            $som = ($som OR 1);
        }
    }
}
?>
<table align=center width=1300 height=125 bgcolor=#44DAAA>

    <tr height = 12% >
        <td align=center colspan="8"><h2>Leituras
Analíticas</h2></td>
    </tr>
    <tr height = 88%>
        <?php echo $an1 ?>
        <?php echo $an2 ?>
        <?php echo $an3 ?>
        <?php echo $an4 ?>
        <?php echo $an5 ?>
        <?php echo $an6 ?>
        <?php echo $an7 ?>
        <?php echo $an8 ?>
    </tr></table>

<br>
<table align="center" width=1300 height=125 bgcolor=#44DAAA>

    <tr height = 12% >
        <td align=center colspan="8"><h2>Leituras Digitais</h2>

```

```

        ></td>
    </tr>
    <tr height = 88%>
        <?php echo $dg1 ?>
        <?php echo $dg2 ?>
        <?php echo $dg3 ?>
        <?php echo $dg4 ?>
        <?php echo $dg5 ?>
        <?php echo $dg6 ?>
        <?php echo $dg7 ?>
        <?php echo $dg8 ?>
    </tr>
</table>
</div>

<div class="rodape">
<!-- <nav class="navbar navbar-default">
    <div class="container-fluid">
        <div class="navbar-header">
            <a class="navbar-brand" href="#">telesite</a>
        </div>
        <ul class="nav navbar-brand">
            <li class="active"><a
href="configura.php">Configurações</a></li>
            <li><a href="alarmes.php">Alarmes</a></li>
        </ul>
    </div>
</nav>-->
    <a href="configura.php" class="button">
Configuração</a>
    <a href="alarmes.php" class="button">Alarmes</a>

    <?php
    if ($som == 1){
        echo $sonoro;}
    ?>
</div>

<div class="rodape">

    <p class= "text">
        <label align = "right">UTFPR - Universidade Tecnológica
Federal do Paraná - Curso Superior de Tecnologia em Sistemas de
Telecomunicações</label><br>
        <label align = "center">
            <?php
            setlocale(LC_TIME, 'pt_BR', 'pt_BR.utf-
8','pt_BR.utf-8', 'portuguese' );
            date_default_timezone_set('America/São Paulo');
            echo strftime('%A, %d de %B de %Y',
strtotime('today')); ?> </label>
        </p>

</div><!--fecha div rodape2-->
</body>
</html>

```

APÊNDICE C - Código Fonte Tela configura.php

```

<?php

$message = "";
include "conndb.php";

if (isset($_POST['submit_anl'])) ){

    $medida = $_POST['medida'];
    $nome = $_POST['nome'];
    $minimo = $_POST['minimo'];
    $maximo = $_POST['maximo'];
    $unidade = $_POST['unidade'];
    $almMin = $_POST['almMin'];
    $almMax = $_POST['almMax'];
    $email = $_POST['email'];
    $mensagem = $_POST['mensagem'];

    $updateQuery = "UPDATE parametros SET nome = '$nome', minimo =
'$minimo', maximo = '$maximo', unidade = '$unidade', almMin =
'$almMin', almMax = '$almMax',email = '$email', mensagem =
'$mensagem' WHERE medida='$medida' ";

    if ($db->exec($updateQuery)){
        $message = "Dados alterados com Sucesso!";}
        else{
            $message = "Dados não Alterados!";
        }
    }

    if (isset($_POST['submit_dig'])) ){

        $medidadig = $_POST['medida'];
        $nome = $_POST['nome'];
        $statusNormal = $_POST['statusNormal'];
        $statusAlarme = $_POST['statusAlarme'];
        $alarmeVal = $_POST['alarmeVal'];
        $email = $_POST['email'];
        $mensagem = $_POST['mensagem'];

        $updateQuery = "UPDATE parametrosdig SET nome = '$nome',
statusNormal = '$statusNormal', statusAlarme = '$statusAlarme',
alarmeVal = '$alarmeVal', email = '$email', mensagem = '$mensagem'
WHERE medidadig='$medidadig' ";

        if ($db->exec($updateQuery)){
            $message = "Dados digitais alterados com Sucesso!";}
            else{
                $message = "Dados digitais não alterados!";
            }
        }

        if (isset($_POST['submit_data_2'])) ){

```

```

        $medida = $_POST['medida'];

    }
    $query = "SELECT rowid, * FROM parametros WHERE medida='$medida'";
    $result = $db->query($query);

?>

<?php

$message = "";

include "conndb.php";

?>

<!DOCTYPE html>

<html>
<head>
    <title> Configuração de Parâmetros de Leitura </title>

    <meta http-equiv="Content-Type" content="html/text";
    charset="utf-8"/>
    <link rel="stylesheet" href="style.css" media="all" />
    <script
src="//ajax.googleapis.com/ajax/libs/jquery/1.11.1/jquery.min.js"></
script>
</head>
<body>

<!-- script para abas de configurações-->
    <script>
function abre_config(evt, aba) {
    var i, tabcontent, tablinks;
    tabcontent = document.getElementsByClassName("tabcontent");
    for (i = 0; i < tabcontent.length; i++) {
        tabcontent[i].style.display = "none";
    }
    tablinks = document.getElementsByClassName("tablinks");
    for (i = 0; i < tablinks.length; i++) {
        tablinks[i].className = tablinks[i].className.replace("
active", "");
    }
    document.getElementById(aba).style.display = "block";
    evt.currentTarget.className += " active";
}
</script>

<div class="superior">
    <h1 align=center>TELESITE</h1><h3 align=center> Sistema de
Gerência de Estações Remotas de TV e Telecom</h3>
</div>

```

```

<hr>
<div class="central"><br>
<div class="a">
    <label for="mensagem">Configuração das Leituras</label>
    <form action ="altera.php" name="medida" method="post"
class="form">

    <p class="medida">
medida:<select name="medida">
    <option value="Analógica_1">Analógica_1</option>
    <option value="Analógica_2">Analógica_2</option>
    <option value="Analógica_3">Analógica_3</option>
    <option value="Analógica_4">Analógica_4</option>
    <option value="Analógica_5">Analógica_5</option>
    <option value="Analógica_6">Analógica_6</option>
    <option value="Analógica_7">Analógica_7</option>
    <option value="Analógica_8">Analógica_8</option>
    <option value="Digital_1">Digital_1</option>
    <option value="Digital_2">Digital_2</option>
    <option value="Digital_3">Digital_3</option>
    <option value="Digital_4">Digital_4</option>
    <option value="Digital_5">Digital_5</option>
    <option value="Digital_6">Digital_6</option>
    <option value="Digital_7">Digital_7</option>
    <option
value="Digital_8">Digital_8</option></select>

    <p class="submit">

    <input name="submit_data_2" type="submit" value="Buscar"
/>

    <div><?php echo $message;?></div><br>

    </form>
</div> <!-- div a-->
<div class="b">

<div class="tab">
    <button class="tablinks" onclick="abre_config(event,
'Analogicas')">Analógicas</button>
    <button class="tablinks" onclick="abre_config(event,
'Digitais')">Digitais</button>
</div>

<div id="Analogicas" class="tabcontent">

<?php
    include('conndb.php');
    $col=$db->query("SELECT medida FROM parametros");
?>

```

```

<table>
  <tr><td>Medida</td><td>Nome</td><td>Mínimo</td><td>Máximo</td><td>Unidade</td><td>Alarme Mínimo</td><td>Alarme Máximo</td><td>e-mail</td><td>Mensagem</td></tr>
<?php
  while($colMedida = $col->fetchArray()){
    $a=$colMedida['medida'];
    $lin=$db->query("SELECT * FROM parametros WHERE
medida=('$a')");
    while($linha = $lin->fetchArray()){
      echo("<tr><td>". $linha ['medida']. "</td><td>".
$linha ['nome']. "</td><td>". $linha ['minimo']. "</td><td>". $linha
['maximo']. "</td><td>". $linha ['unidade']. "</td><td>". $linha
['almMin']. "</td><td>". $linha ['almMax']. "</td><td>". $linha
['email']. "</td><td>". $linha ['mensagem']. "</td></tr>");
    }
  }
?>
</table>

```

```
</div><!-- div tab analógicas-->
```

```
<div id="Digitais" class="tabcontent">
```

```

<?php
include('conndb.php');

$col=$db->query("SELECT medidadig FROM parametrosdig");?>

<table>
<tr><td>Medida</td><td>Nome</td><td>Status Normal</td><td>Status
Alarme</td><td>Valor de Alarme</td><td>e-mail</td><td>Mensagem</td></tr>
<?php
while($colMedida = $col->fetchArray()){
  $b=$colMedida['medidadig'];

  $lin=$db->query("SELECT * FROM parametrosdig WHERE
medidadig=('$b')");
  while($linha = $lin->fetchArray()){

    echo("<tr><td>". $linha ['medidadig']. "</td><td>".
$linha ['nome']. "</td><td>". $linha ['statusNormal']. "</td><td>".
$linha ['statusAlarme']. "</td><td>". $linha ['alarmeVal'].
"</td><td>". $linha ['email']. "</td><td>". $linha ['mensagem'].
"</td></tr>");
  }
}

?>
</table>

```

```
</div><!-- div tab digitais-->
</div><!-- div b-->

</div>
</p>
<a
href="inicial.php"class="button">Voltar</a>
</div>
<hr>

<div class="footer">

    <p class= "text">
        <label align = "right">Sistema de Gerência de Estações
Remotas de TV e Telecom</label><br>
        <label align = "center"> <?php echo date('d-m-Y
H:i');?></label>
    </p>

</div><!-- fecha div rodape-->

</body>
</html>
```

APÊNDICE D - Código Fonte Tela altera.php

```

<?php
$message = "";
include "conndb.php";

if (isset($_POST['submit_data_2']) ){
    $medida = $_POST['medida'];
}
?>

<!DOCTYPE html>

<html>
<head>
    <title>UTFPR Sistel - TCC</title>
    <meta http-equiv="Content-Type" content="html/text";
charset="utf-8"/>
    <link rel="stylesheet" href="style.css" media="all" />
    <script
src="//ajax.googleapis.com/ajax/libs/jquery/1.11.1/jquery.min.js"></
script>
</head>
<body>
    <div class="superior">
        <h1 align=center>TELESITE</h1><h3 align=center> Sistema
de Gerência de Estações Remotas de TV e Telecom</h3>
    </div>
<hr>
<div class="central">
<br>
    <?php
        if ((strlen($medida))==12){/* abre formulário de medidas
analógicas*/

            /**
                $col=$db->query("SELECT medida FROM parametros");
                ($colMedida = $col->fetchArray()){
                $a=$colMedida['medida'];
                $lin=$db->query("SELECT * FROM parametros WHERE
medida=('$a')");
                while($linha = $lin->fetchArray()){
                    echo("<tr><td>". $linha ['medida']. "</td><td>".
$linha ['nome']. "</td><td>". $linha ['minimo']. "</td><td>". $linha
['maximo']. "</td><td>". $linha ['unidade']. "</td><td>". $linha
['almMin']. "</td><td>". $linha ['almMax']. "</td><td>". $linha
['email']. "</td><td>". $linha ['mensagem']. "</td></tr>");
                }
                * $query = "SELECT rowid, * FROM parametros WHERE
medida='$medida'";
                * $qSelect = "SELECT * FROM parametros WHERE
medida=('Analógica_1')";
                $result=$db->query($qSelect);

```

```

while($row1 = $result->fetchArray()){
    *
    *
    **/

    $querySel = "SELECT rowid, * FROM parametros WHERE
medida='$medida'";
    $result = $db->query($querySel);
    while($row = $result->fetchArray()) {?>

        <form action ="configura.php" name="medida" method="post"
class="form">
            <table>
                <tr>
                    <td colspan = "3">
                        <p class="medida">
                            <label for="medida">Alterar: </label>
                            <input type="medida" name="medida" value=<?php
echo ($row['medida']);?> />
                        </td>
                    </tr>
                    <tr>
                        <td>
                            <p class="nome">
                                <label for="nome">Nome da Leitura</label>
                                <input type="nome" name="nome" value=<?php echo
$row['nome'];?> />
                            </td>
                        <td>
                            <p class="minimo">
                                <label for="minimo">Valor m nimo</label>
                                <input type="minimo" name="minimo" value=<?php
echo $row['minimo'];?> />
                            </td>
                        <td>
                            <p class="maximo">
                                <label for="maximo">Valor m ximo</label>
                                <input type="maximo" name="maximo" value= <?php
echo $row['maximo'];?> />
                            </td>
                        </tr>
                    <tr>
                        <td>
                            <p class="unidade">
                                <label for="unidade">Unidade</label>
                                <input type="unidade" name="unidade"
value=<?php echo $row['unidade'];?> />
                            </td>
                        <td>
                            <p class="almMin">
                                <label for="almMin">Alarme Inferior</label>
                                <input type="almMin" name="almMin" value=<?php
echo $row['almMin'];?> />
                            </td>
                        <td>
                            <p class="almMax">

```

```

        <label for="almMax">Alarme Superior</label>
        <input type="almMax" name="almMax" value=<?php
echo $row['almMax'];?> />
    </td>
</tr>
<tr>
    <td>
        <p class="email">
        <label for="email">E-mail</label>
        <input type="email" name="email" value=<?php
echo $row['email'];?> />
    </td>
    <td colspan = "2">
        <p class="mensagem">
        <label for="mensagem">Mensagem de Aviso</label>
        <textarea name="mensagem" value=<?php echo
($row['mensagem']);?> />
        </textarea>
    </td>
</tr> </table><table align=center>
<tr>
    <!--<td align = "right" colspan = "3"--><td>
        <p class="submit">
        <input type="submit" name="btnCancela"
value="Cancela" onclick="form.action='inicial.php';form.submit()"
/></td><td>
        <p class="submit">
        <input name="submit_anl" type="submit"
value="Salvar" />
    </td>
</tr>
</table>
</form>

<?php }}else{ /* abre formulário de medidas digitais*/

    $query = "SELECT rowid, * FROM parametrosdig WHERE
medidadig='$medida'";
    $result = $db->query($query);
    while($row = $result->fetchArray()) {?>

        <form action ="configura.php" name="medida" method="post"
class="form">
        <table>
            <tr>
                <td colspan = "3">
                    <p class="medida">
                    <label for="medida">Alterar: </label>
                    <input type="medida" name="medida" value=<?php
echo $row['medidadig'];?> />
                </td>
            </tr>
            <tr>
                <td>

```

```

        <p class="nome">
        <label for="nome">Nome da Leitura</label>
        <input type="nome" name="nome" value=<?php echo
$row['nome'];?> />
    </td>
    <td>
        <p class="statusNormal">
        <label for="statusNormal">SituaçÃo
Normal</label>
        <input type="statusNormal" name="statusNormal"
value=<?php echo $row['statusNormal'];?> />
    </td>
    <td>
        <p class="statusAlarme">
        <label for="statusAlarme">SituaçÃo
Alarme</label>
        <input type="statusAlarme" name="statusAlarme"
value=<?php echo $row['statusAlarme'];?> />
    </td>
</tr>
<tr>
    <td>
        <p class="Valor de alarme">
        <label for="alarmeVal">Valor de Alarme</label>
        <input type="alarmeVal" name="alarmeVal"
value=<?php echo $row['alarmeVal'];?> />
    </td>
    <td>
    </td>
    <td>
    </td>
</tr>
<tr>
    <td>
        <p class="email">
        <label for="email">E-mail</label>
        <input type="email" name="email" value=<?php
echo $row['email'];?> />
    </td>
    <td colspan = "2">
        <p class="mensagem">
        <label for="mensagem">Mensagem de Aviso</label>
        <textarea name="mensagem" value=<?php echo
$row['mensagem'];?> />
    </td>
</tr> </table><table align=center>
<tr>
    <!--<td align = "right" colspan = "3">--><td>
        <p class="submit">
        <input type="submit" name="btnCancela"
value="Cancela" onclick="form.action='inicial.php';form.submit()"
/></td><td>
        <p class="submit">

```

```

                                <input name="submit_dig" type="submit"
value="Salvar" />
                                </td>
                            </tr>
                        </table>
                    </form>

                                <?php }} ?>

</div>

<div class="rodape2">

    <p class= "text"><br>
        <label align = "right">UTFPR - Universidade Tecnológica
Federal do Paraná - Curso Superior de Tecnologia em Sistemas de
Telecomunicações</label><br>

        <!--<label align = "center"> <?php echo
date('H:I');?></label>-->
        </p>

</div><!--fecha div rodape2-->

</body>
</html>
```

APÊNDICE E - Código Fonte Arquivo de Estilos style.css

```
body {
    padding: 00px 70px;
    font-size: 22px;
    font-family: arial, Tahoma, sans-serif;
    background: linear-gradient( #DDDDEE, #ADD8E6);
}

.tab {
    overflow: hidden;
    border: 1px solid #ccc;
    background-color: #C1f1f5;
}

.tab button {
    background-color: inherit;
    float: left;
    border: none;
    outline: none;
    cursor: pointer;
    padding: 14px 16px;
    transition: 0.3s;
    font-size: 17px;
}

.tab button:hover {
    background-color: #ddd;
}

.tab button.active {
    background-color: #ccc;
}

.tabcontent {
    display: none;
    padding: 6px 12px;
    border: 1px solid #ccc;
    border-top: none;
}

div.container{
    height:100%;
    background:-webkit-linear-gradient(top, #088EBC 20%, #004A67
100%);
}

div.superior{
    width: 1350px;
```

```
        height:110px;
        background:-webkit-linear-gradient(left, #009FCD 70%, #006A97
100%);
        border-radius:3px;
    }
    div.central{
        width: 1350px;
        height:400px;
        background:-webkit-linear-gradient(left, #088EBC 70%, #006A97
100%);
        border-radius:3px;
    }

    div.rodape{

        width: 1350px;
        height:60px;
        background:-webkit-linear-gradient(left, #088EBC 70%, #006A97
100%);
        border-radius:3px;
    }

    div.rodape2{
        width: 1350px;
        height:60px;

        background:-webkit-linear-gradient(left, #088EBC 70%, #006A97
100%);
        border-radius:3px;
    }
    div.a{
        width: 25%;
        float:left;
        position: relative;
    }

    div.b{
        width: 75%;
        float:right;
        position: relative;
    }

    #header{
        width:1300px;
        height:50px;
        margin-top:0;
        margin-right:auto;
        margin-left:auto;
        margin-bottom:0;
        background-color:#0A2A29;
        border: 1px solid #E5E5E5;

    }
```

```
#central{
    width:1300px;
    height:200px;
    margin-top:100;
    margin-right:auto;
    margin-left:auto;
    margin-bottom:0;
    background-color:#EFFBFB;
    border: 1px solid #E5E5E5;
}

.btn_nome {
    border: 1px;
    color: black;
    padding: 1px 5px;
    cursor: pointer;
    font-size: 16px;
    border-radius:5px;
    height: 90px;
    width:140px;
    background-color: #0000FF;
}

.btn_nome:hover {
    background: #0044FF;
    color: blue;
    font: bold;
    font-size: 16px;
}

.btn_normal {
    border: 1px;
    color: black;
    padding: 1px 5px;
    cursor: pointer;
    font-size: 16px;
    border-radius:5px;
    height: 90px;
    width:150px;
    background-color: #00FF77;
}

.btn_normal:hover {
    background: #00FF33;
    color: black;
    font-size: 16px;
    height: 90px;
    width:150px;
}

.btn_alarm {
    border: none;
    color: black;
    padding: 1px 5px;
    cursor: pointer;
    font-size: 16px;
}
```

```
border-radius:5px;
height: 90px;
width:150px;
background-color: #FF0000;
}
.btn_alarm:hover {
background: #DD0000;
color: black;
font-size: 16px;
height: 90px;
width:150px;
}
a { color:#123456; }

h1 {
margin-bottom: 10px;
color: white;

text-shadow:3px 3px 4px #000000;
}

h2 {
margin-top: 10px;
margin-bottom: 10px;
color: black;
}

h3 {
margin-bottom: 10px;
color: white;

text-shadow:2px 2px 4px #0000DD;
-webkit-transition: width 4s;
}
text {
margin-bottom: 10px;
color: white;
text-shadow:2px 2px 4px #0000DD;
-webkit-transition: width 4s;
}

input, textarea {
padding: 6px;
border: 1px solid #E5E5E5;
width: 200px;
color: #666666;
box-shadow: rgba(0, 0, 0, 0.1) 0px 0px 8px;
-moz-box-shadow: rgba(0, 0, 0, 0.1) 0px 0px 8px;
-webkit-box-shadow: rgba(0, 0, 0, 0.1) 0px 0px 8px;
}

textarea {
width: 200px;
```

```

        height:40px;
        max-width: 400px;
        line-height: 18px;
    }

input:hover, select:hover, td:hover, select:hover,h2:hover,
textarea:hover,
input:focus, select:focus, td:focus, select:focus,h2:focus,
textarea:focus {
    border-color: 1px solid #C9C9C9;
    box-shadow: rgba(0, 0, 0, 0.2) 0px 0px 8px;
    -moz-box-shadow: rgba(0, 0, 0, 0.2) 0px 0px 8px;
    -webkit-box-shadow: rgba(0, 0, 0, 0.2) 0px 0px 8px;
}

form label {
    border:1px;
    margin-bottom: 1px;
    color: #444444;
    display: inline;
}
label2 {
    margin-bottom: 2px;
    color: #444444;

    display: inline;
}

table {
    border:1px;
    border-color:#1537cc;
    border-collapse: separate;
    background-color:#DDDDFF;
    font-size: 16px;
    position: static;
    resize: none;
    border-radius:5px;
}

table2 {
    border:2px;
    border-color:#1537cc;
    border-collapse: separate;
    background-color:#DDDDFF;
    font-size: 16px;
    position: static;
    resize: none;
    border-radius:5px;
}

tr:ntn-child(odd){
background-color:#DDDDDD;
}

td{

```

```
padding:2px;
color: black;
border:1px;
border-color:#153781;
border-collapse: separate;
/*background-color:#DDDDDD;*/
border-radius:1px;
}

select {
width:220px;
margin-bottom: 5px;
color: #999999;
display: inline-block;
}

.submit input {
width: 100px;
height: 40px;
background-color: #1122FF;
color: #FFFFFF;
border-radius: 6px;
-moz-border-radius: 5px;
-webkit-border-radius: 3px;
}

.button {
width: 130px;
height: 60px;

background-color: #2222FF;
color: #AAAAAA;
border-radius: 3px;
-moz-border-radius: 5px;
-webkit-border-radius: 3px;
}
```

ANEXO A - Folha de Dados do CI MCP3008



MCP3004/3008

2.7V 4-Channel/8-Channel 10-Bit A/D Converters with SPI Serial Interface

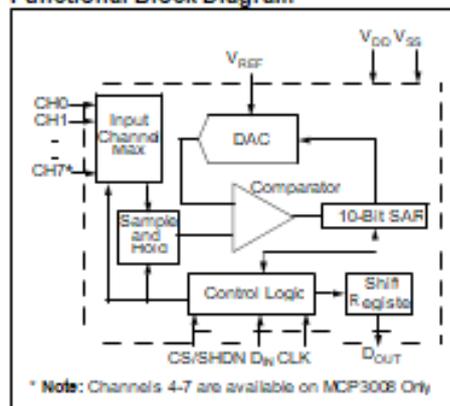
Features

- 10-bit resolution
- ± 1 LSB max DNL
- ± 1 LSB max INL
- 4 (MCP3004) or 8 (MCP3008) input channels
- Analog inputs programmable as single-ended or pseudo-differential pairs
- On-chip sample and hold
- SPI serial interface (modes 0,0 and 1,1)
- Single supply operation: 2.7V - 5.5V
- 200 ksp/s max. sampling rate at $V_{DD} = 5V$
- 75 ksp/s max. sampling rate at $V_{DD} = 2.7V$
- Low power CMOS technology
- 5 nA typical standby current, 2 μ A max.
- 500 μ A max. active current at 5V
- Industrial temp range: $-40^{\circ}C$ to $+85^{\circ}C$
- Available in PDIP, SOIC and TSSOP packages

Applications

- Sensor Interface
- Process Control
- Data Acquisition
- Battery Operated Systems

Functional Block Diagram



Description

The Microchip Technology Inc. MCP3004/3008 devices are successive approximation 10-bit Analog-to-Digital (A/D) converters with on-board sample and hold circuitry. The MCP3004 is programmable to provide two pseudo-differential input pairs or four single-ended inputs. The MCP3008 is programmable to provide four pseudo-differential input pairs or eight single-ended inputs. Differential Nonlinearity (DNL) and Integral Nonlinearity (INL) are specified at ± 1 LSB. Communication with the devices is accomplished using a simple serial interface compatible with the SPI protocol. The devices are capable of conversion rates of up to 200 ksp/s. The MCP3004/3008 devices operate over a broad voltage range (2.7V - 5.5V). Low-current design permits operation with typical standby currents of only 5 nA and typical active currents of 320 μ A. The MCP3004 is offered in 14-pin PDIP, 16-pin SOIC and TSSOP packages, while the MCP3008 is offered in 16-pin PDIP and SOIC packages.

Package Types

