UNIVERSIDADE FEDERAL DE SANTA MARIA
COLÉGIO AGRÍCOLA DE FREDERICO WESTPHALEN
ESPECIALIZAÇÃO LATO-SENSU EM GESTÃO DE TECNOLOGIA DA INFORMAÇÃO

SOLUÇÃO PARA MONITORAMENTO E REGISTRO DE RECURSOS DE TI EM SALAS DE SERVIDORES, UTILIZANDO ARDUINO

MONOGRAFIA DE ESPECIALIZAÇÃO

Jean Pablo Dreyer

Frederico Westphalen, RS, Brasil

2013

2013	Especialista	DREYER, Jean P.	PG-E GTI/UFSM, RS

SOLUÇÃO PARA MONITORAMENTO E REGISTRO DE RECURSOS DE TI EM SALAS DE SERVIDORES, UTILIZANDO ARDUINO

por

Jean Pablo Dreyer

Monografia apresentada ao Curso de Pós-Graduação Lato-Sensu em Gestão da Tecnologia da Informação, da Universidade Federal de Santa Maria, Colégio Agrícola de Frederico Westphalen (UFSM, CAFW; RS), como requisito parcial para a obtenção do título de
Especialista em Gestão da Tecnologia da Informação

Orientador: Prof. Mestre. Evandro Preuss

Frederico Westphalen, RS, Brasil

2013

Universidade Federal de Santa Maria

Colégio Agrícola de Frederico Westphalen
Pós-graduação em Gestão de Tecnologia da Informação

A Comissão Examinadora, abaixo assinada,
aprova a Monografia de Especialização

SOLUÇÃO PARA MONITORAMENTO E REGISTRO DE RECURSOS DE TI EM SALAS DE SERVIDORES, UTILIZANDO ARDUINO

elaborado por:
Jean Pablo Dreyer

como requisito parcial para a obtenção de título de
Especialista em Gestão da Tecnologia da Informação

COMISSÃO EXAMINADORA

Evandro Preuss, Ms.(PUC/RS)
(Presidente Orientador)

Roberto Franciscatto, Ms.(UNISINOS)

Bruno Batista Boniati, Ms.(UFSM)

Frederico Westphalen, 02 de dezembro de 2013.

AGRADECIMENTOS

A Deus, por não ter permitido que eu fraquejasse durante esta etapa.

Ao professor e amigo Evandro Preuss, orientador da presente monografia, pelo apoio, orientação e disponibilidade.

A Patrícia, minha esposa, por toda a compreensão, preocupação, dedicação, incentivo e enorme ajuda a ultrapassar os obstáculos encontrados.

Aos demais professores, que dedicaram seu tempo para realizar este curso, em especial ao professor Roberto Franciscatto.

A UNIVERSIDADE FEDERAL DE SANTA MARIA, COLÉGIO AGRÍCOLA DE FREDERICO WESTPHALEN, PELA OPORTUNIDADE.

RESUMO

Monografia de Especialização
Programa de Pós-Graduação em Gestão de Tecnologia da Informação
Universidade Federal de Santa Maria

SOLUÇÃO PARA MONITORAMENTO E REGISTRO DE RECURSOS DE TI EM SALAS DE SERVIDORES, UTILIZANDO ARDUINO.

AUTOR: JEAN PABLO DREYER.
ORIENTADOR:EVANDRO PREUSS.
Data e Local da Defesa: Frederico Westphalen, dezembro de 2013.

Negócios de hoje e do futuro exigem a total disponibilidade dos serviços de tecnologia da informação (TI). O presente trabalho tem por objetivo o estudo do framework CobiT e da norma ABNT NBR ISO/IEC 17799, a fim de identificar os conceitos associados à governança de TI, relacionando-os com a gestão da continuidade dos serviços. Também é objetivo apresentar uma solução para monitoramento e registros de recursos de TI em salas de servidores, utilizando hardware e software, que seja de fácil personalização, onde se aplicam os conceitos das normas, fazendo com que o produto desenvolvido emita avisos sonoros e/ou visuais, envie e-mails e SMS de alertas, e registre todas as anomalias no ambiente físico. Isso pode ajudar os administradores a tomarem providências antes que a continuidade do serviço seja interrompida por falhas, como por exemplo, superaquecimento, fumaça, falta de energia, umidade, entre outros. O hardware utilizado é o Arduino, que é uma plataforma eletrônica de hardware livre, projetada com um microcontrolador de placa única, com suporte de entrada e saída embutido, com uma linguagem de programação padrão e com essência em C/C++.

Palavras-chave: Arduino, Gestão de TI, CobiT, ISO 17799, microcontrolador.

ABSTRACT

Specialization Monograph
Postgraduate Program in Management of Information Technology

LOW COST SOLUTION FOR MONITORING AND REGISTRATION OF APPEALS OF IT IN HALLS OF SERVERS USING ARDUINO.
AUTHOR: JEAN PABLO DREYER.
ADVISOR: EVANDRO PREUSS.
Date and Defense Local: Frederico Westphalen, December 2013.

Business today and in the future require the full availability of services for information technology (IT). The present work aims to study the frameworks COBIT and ISO / IEC 17799 in order to identify the concepts associated with IT governance, linking them with the continuity management services. It is also intended to present a solution for monitoring and resource records IT server rooms, using hardware and software that is easily customized, which apply the concepts of standards, making the product developed emit beeps and / or visual, send emails and SMS alerts, and record any deficiencies in the physical environment of the data center. This can help administrators to take action before the service continuity is interrupted by faults, such as overheating, smoke, lack of energy, moisture, among others. The hardware used is the Arduino, which is an electronic platform of free hardware, designed with a single-board microcontroller, supported input and output built with a standard programming language and essence in C / C + +.

Keywords: Arduino, IT management, CobiT, ISO 17799, microcontroller.

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

ABNT - Associação Brasileira de Normas Técnicas.

BD – Banco de Dados.

C/C++ - Linguagem de Programação.

COBIT - Control Objectives for Information and related Technology.

CPD - Centros de Processamentos de Dados.

CSS - Cascading Style Sheets.

EEPROM - Electrically-Erasable Programmable Read-Only Memory.

EPC - Electronic Product Code.

ER – Entidade Relacionamento.

ERP - Enterprise Resource Planning.

GHz - Unidade de Frequência.

Gnd - Abreviação de aterramento em placas eletrônicas.

GPL - General Public License.

GPRS - General Packet Radio Services.

GPS - Global Positioning System.

GSM - Global System for Mobile Communications.

HTML - HyperText Markup Language.

HTTP - Hypertext Transfer Protocol.

ICSP - In System Chip Programming.

IDE - Integrated Development Environment.

IEC - International Electrotechnical Commission.

IP - Protocolo de comunicação de rede.

ISO - International Organization for Standardization.

KB – Unidade de Medida de Dados. 1 KB = 1.024 bytes.

kHz - Unidade de Frequência.

KΩ – Abreviatura de quilo-ohm.

LCD - Liquid Crystal Display.

LED - Light Emitting Diode.

mA – Abreviatura de Miliampere.

MAC - Media Access Control.

MHz - Unidade de Frequência.

MP3 – Formato de arquivos de áudio.

MySQL – Sistema gerenciador de Banco de Dados.

NBR – Abreviatura para Norma Brasileira.

PC – Personal Computer.

PHP - Acrónimo recursivo para "PHP: Hypertext Preprocessor".

PWM – PulseWidth Modulation.

RFID – Identificação por Rádio Frequência.

RGB - Abreviatura do sistema de cores aditivas formado por Vermelho, Verde e Azul.

RJ45 – Padrão de conector de redes de computadores.

SQL - Structured Query Language.

SRAM - Static Random Access Memory.

TI – Tecnologia da Informação.

UART - Universal Asynchronous Receiver-Transmitter.

USB - Universal Serial Bus.

V – Volts.

Vcc – Abreviatura de pino de alimentação em placas eletrônicas.

WEB - World Wide Web (que em português significa, "Rede de alcance mundial").

SUMÁRIO

1 INTRODUÇÃO

A competição com novos concorrentes, o surgimento de produtos e serviços substitutos, os clientes mais conscientes, a exigência de maior transparência e dinamismo nos negócios e a maior velocidade de comunicação são algumas características que fizeram com que a Tecnologia da Informação se tornasse vital para que empresas alcancem o sucesso.

A Tecnologia da Informação permite criar uma infinidade de aplicações que podem auxiliar, aprimorar e facilitar desde a automatização de tarefas até a tomada de decisões gerenciais. Empresas e organizações dependem destas aplicações para exercer suas funções chave, tais como, produção, vendas, finanças e *marketing*.

Assim, os sistemas de informação além de serem conhecidos pelos benefícios que trazem para a gestão dos negócios, no qual se tenta eliminar desperdícios e tarefas demasiadamente repetitivas, programar melhorias nos controles dos custos e da qualidade de um produto ou serviço, podem vir a se transformar num ponto crítico para o insucesso de qualquer empresa, caso venha a ficar inoperante. Falhas nos sistemas e na gestão da TI podem gerar transtornos irreversíveis, como prejuízos financeiros e manchas no nome da empresa.

Todas as empresas e organizações que utilizam algum sistema de TI estão relacionadas diretamente ou indiretamente com Centros de Processamentos de Dados (CPDs). Fatores ambientais, tais como proteção contra incêndios, energia elétrica, calor e umidades excessivas, resíduos, sujeiras ou poeiras, devem ser monitorados e controlados, através de planos de contingência, para minimizar o risco e a interrupção desnecessária dos serviços e negócios.

Para tanto, é preciso ter planos que garantam que as falhas não gerem problemas e/ou caso eles aconteçam sejam sanados o mais rápido possível. Em todo esse processo de gestão, torna-se necessário um processo que registre esses incidentes, pois são inúmeras variáveis que precisam ser gerenciadas e, geralmente, estes incidentes não são gerenciados, tanto menos registrados, acarretando inúmeras dificuldades para gestão.

Um dos problemas encontrados pelas empresas é a falta de conhecimento e consequentemente a falta da aplicação das normas de TI. O *framework* para Governança de TI

– COBIT – trata dessa dificuldade no item “Disponibilidade”, a qual relaciona-se com a disponibilidade da informação quando exigida pelo processo de negócio hoje e no futuro. Também está ligada à preservação dos recursos necessários e capacidades associadas.

Atualmente o mercado apresenta uma carência de soluções completas e personalizadas, disponibilizando somente soluções específicas que não podem ser personalizadas, de custo elevado e que muitas vezes não atendem as reais necessidades da empresa.

Este trabalho apresenta um estudo e proposta de uma solução que ajude a manter a continuidade dos *data centers*, através do uso da plataforma eletrônica de hardware livre Arduino, fornecendo soluções de monitoramento aos fatores ambientais citados acima, controlando e monitorando cuidadosamente o acesso, evitando que pessoas sem permissão tenham acesso aos equipamentos.

O segundo capítulo apresenta os conceitos associados à governança de TI, relacionando-os à gestão da tecnologia da informação e à gestão da continuidade dos serviços, a fim de que as empresas consigam alinhar estrategicamente o gerenciamento dos riscos de TI, sem prejudicar os seus negócios. O capítulo 3 apresenta o Arduino, que fará parte da solução proposta por este trabalho, O capítulo 4 apresenta os trabalhos relacionados e o quinto capítulo apresenta a proposta e o desenvolvimento da aplicação de monitoramento e registro de recursos de ti em salas de servidores, utilizando Arduino.

2 GOVERNANÇA DE TI

Gaseta (2012) destaca que a Governança de TI é uma prática que as organizações públicas e privadas procuram implementar com o objetivo de prover mais controle da estrutura de TI, procurando garantir sustentabilidade e competitividade, por meio de alinhamento entre os objetivos da TI e os objetivos estratégicos da organização.

No mesmo sentido, ensina Fernandes (2012) que o principal objetivo da Governança de TI é apurar a TI às condições necessárias para se alcançar o negócio, considerando respostas de apoio ao negócio, assim como a promessa da continuidade dos serviços e a minimização da exposição do negócio aos riscos de TI.

Examinando o objetivo principal da TI, como acima exposto, podemos identificar, ainda, em uma análise detalhada, segundo Fernandes (2012), outros objetivos de Governança de TI:

- Promover o posicionamento mais claro e consistente da TI em relação às demais áreas de negócio da empresa;
- Promover o alinhamento e a priorização das iniciativas de TI com a estratégia de negócio;
- Promover o alinhamento da arquitetura de TI, sua infraestrutura e aplicações às necessidades do negócio, em termos de presente e futuro;
- Promover a implantação e melhoria dos processos operacionais e de gestão necessários para atender aos serviços de TI, conforme padrões que atendam às necessidades do negócio;
- Prover a TI da estrutura de processos que possibilite a gestão do seu risco e o cumprimento das regras e normas, para a continuidade operacional da empresa;
- Promover o emprego de regras claras para as responsabilidades sobre decisões e ações relativas à TI, no âmbito da empresa;

Por outra vertente, é importante mencionar, que a visão de Governança de TI pode ser representada pelo que Fernandes (2012) chama de “Ciclo de Governança de TI”, formado por quatro grandes etapas: alinhamento estratégico e *compliance*,

decisão, estrutura e processos e gestão do valor e do desempenho, consoante se pode verificar na Figura 1.

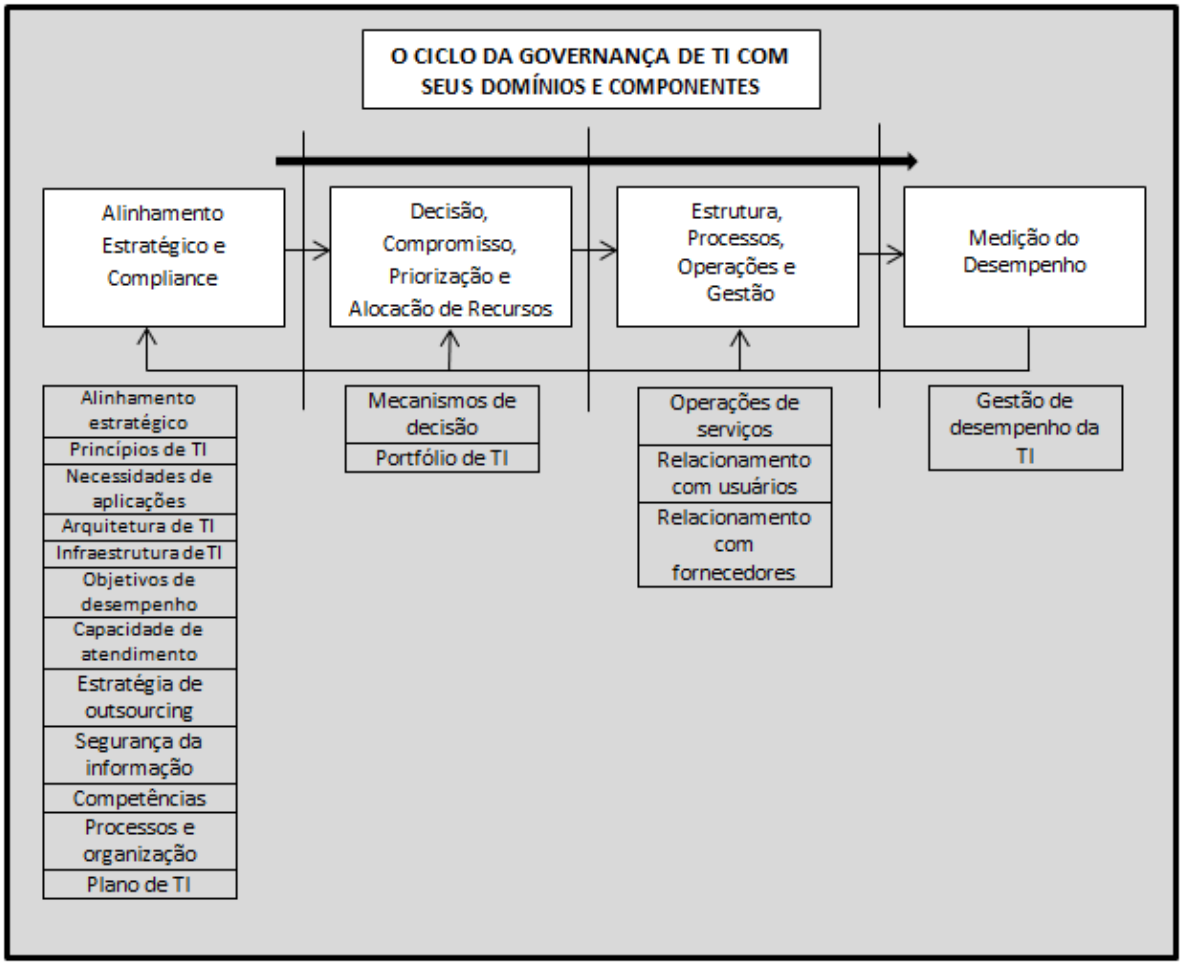

Figura 1 - Ciclo da governança de TI com seus domínios e componentes.

Conclui-se do ciclo demonstrado na figura acima, que a Governança de TI abrange vários mecanismos e componentes que, ao se integrarem, possibilitam o alinhamento estratégico e a continuidade dos negócios.

Portanto, como refere Weill (2006), uma boa governança de TI harmoniza decisões sobre a administração e a utilização da TI com comportamentos desejáveis e objetivos do negócio, visto que sem estruturas de governança cuidadosamente projetadas e implementadas, as empresas deixam essa harmonia ao acaso, o que gera insegurança e prejuízos no campo dos negócios.

2.1 Gestão da continuidade dos serviços

O'brien (2010) diz que um sistema de informação é um conjunto organizado de pessoas, hardware, software, redes de comunicações e recursos de dados, que coleta, transforma e dissemina informações em uma organização.

A informação é o bem mais valioso para muitas empresas e organizações. Porém, a tecnologia que a suporta muitas vezes não recebe o devido valor, sendo mal compreendida. A tecnologia da informação passou a mediar todos os processos operacionais da empresa e, cada vez mais, está afetando a capacidade competitiva, influenciando a posição de custo, a qualidade do serviço aos clientes e a capacidade de inovação.

Organizações bem-sucedidas reconhecem os benefícios da TI e a utiliza para direcionar os valores das partes interessadas no negócio. Além disso, tais empresas entendem os riscos associados tais como as crescentes demandas regulatórias e a dependência crítica de muitos processos de TI. Por isso, a habilidade de uma empresa de alavancar o potencial da tecnologia está sendo reconhecida, pelo mercado e pela alta diretoria, como um fator crítico de sucesso.

A necessidade da avaliação do valor de TI, o gerenciamento dos riscos relacionados à TI e as crescentes necessidades de controle sobre as informações são agora entendidos como elementos-chave da governança corporativa. Valor, risco e controle constituem a essência da governança de TI. COBIT 4.1 (2007)

Até mesmo o mais desavisado dos administradores, aquele que costuma não entender de TI, já percebeu que o não gerenciamento ou o mau gerenciamento da TI é um risco para seu negócio. É provável que ele já tenha passado por um incidente de indisponibilidade ou perda de dados de aplicações que são críticas para a continuidade de seu negócio.

> A governança de TI é de responsabilidade da alta administração (incluindo diretores e executivos), na liderança, nas estruturas organizacionais e nos processos que garantem que a TI da empresa sustente e estenda as estratégias e objetivos da organização (Fernandes, 2012, p. 12).

Como já mencionado alhures, o objetivo da Governança de TI é buscar o alinhamento da TI aos requisitos do negócio, considerando soluções de apoio ao negócio, assim como a garantia da continuidade dos serviços e a redução da exposição do negócio aos riscos de TI.

Assim, fracionando este objetivo principal, pode-se citar outros objetivos: emitir um posicionamento mais claro e consistente em relação às outras áreas da empresa; alinhar e priorizar as iniciativas de TI com a estratégia do negócio; alinhar a arquitetura e a infraestrutura de TI as necessidades do negócio; munir a TI com processos operacionais e de gestão necessários para atender aos serviços de TI.

A Governança de TI habilita a organização a obter todas as vantagens de sua informação, maximizando os benefícios, capitalizando as oportunidades e ganhando em poder competitivo. COBIT 4.1 (2007)

Isso porque, conforme leciona Fernandes (2012), no contexto atual do mercado existem alguns modelos de referência que abordam a TI de forma global, abrangendo os seus princípios de diretrizes tanto no âmbito das organizações de TI quanto do seu relacionamento com as demais organizações que fazem parte da sua cadeia de valor, como por exemplo, clientes, fornecedores e parceiros.

No presente trabalho, destacar-se-ão os seguintes modelos, quais sejam, o CobiT e a norma ABNT NBR ISO/IEC 17799.Além desses, o autor ressalta ainda sobre a norma ISO/IEC 38500, o Val IT e o Risk IT, que, todavia, não são objeto desta pesquisa.

2.1.1 COBIT

O *Control Objectives for Information and related Tecnology* (CobiT) é um *framework*, ou seja, um modelo ou um conjunto de ferramentas que reúne as melhores práticas para a Governança, Segurança e Controle de TI. Estas boas práticas representam um consenso de especialistas e aplicam seu maior foco nos controles e menos na execução, tendo como objetivo ajudar a otimização de investimentos em TI, assegurar a entrega dos serviços e prover métricas para julgar quando as coisas saem erradas.

O modelo CobiT foi criado com as principais características de ser focado em negócios, orientado a processos, baseado em controles e orientado por medições - COBIT 4.1 (2007).

Para atender aos requisitos de negócio para a TI, as empresas devem investir em recursos necessários para criar uma capacidade técnica adequada que resulte no desejado retorno. Por exemplo: um sistema ERP que atenda as necessidades do negócio e que apresente como resultado o aumento de vendas e benefícios financeiros.

Os recursos de TI identificados por este framework são divididos da seguinte maneira:

1) Pessoas: são os funcionários/colaboradores, terceirizados ou contratados, que se fazem necessários para planejar, organizar, adquirir, implementar, entregar, suportar, motivar e avaliar os sistemas de informações e serviços.

2) Infraestrutura: refere-se à tecnologia e aos recursos necessários para possibilitar o processamento dos aplicativos, tais como hardware, sistemas operacionais, sistemas de gerenciamento de bases de dados, redes entre outros.

3) Aplicativos: sistemas automatizados para usuários e também os procedimentos manuais que processam a informação.

4) Informações: são os dados utilizados pelos negócios, dispostos em todas as suas formas, a entrada, o processamento e a saída fornecida pelo sistema de informação em qualquer formato.

Para garantir a eficiência da Governança de TI, é fundamental avaliar as atividades e riscos da TI que precisam ser gerenciados. Geralmente eles são ordenados por domínios de responsabilidade de planejamento, construção, processamento e monitoramento. No modelo, esses domínios são classificados em quatro grupos, que apresentam um inter-relacionamento, conforme a figura 1 e que estão descritos a seguir:

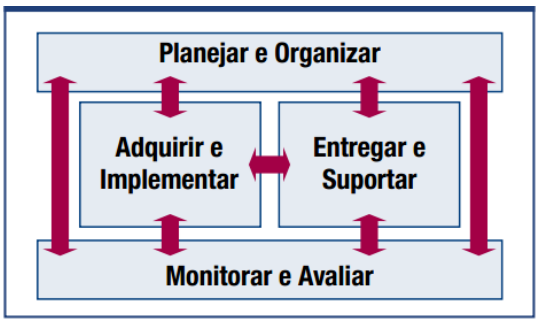

Figura 2 - Classificação dos domínios do CobiT – COBIT 4.1, (2007)

5) Planejamento e Organização: domínio onde são definidos o plano estratégico, a arquitetura das informações e a estrutura de TI.

6) Aquisição e Implementação: domínio onde ocorre a aquisição de softwares aplicativos, infra-estrutura tecnológica e recursos de TI.

7) Distribuição e Suporte: domínio onde são gerenciados níveis de serviço, serviços de terceiros, desempenho e capacidade, atendimento aos clientes, problemas e incidentes, configurações e dados.

8) Monitoração e Avaliação: domínio onde são monitorados e avaliados o desempenho e os controles internos da TI.

Entre os quatro domínios, trinta e quatro processos de TI foram identificados, e possuem uma ligação com os objetivos de negócios e de TI suportados. Podem ser utilizados para verificar a totalidade das atividades e responsabilidades. Porém, nem todos precisam ser aplicados, podendo ser combinados conforme a necessidade de cada empresa.

Além dos quatro domínios principais que guiam o bom uso da tecnologia da informação da organização existe também a questão de auditoria que permite verificar, através de relatórios de avaliação, o nível de maturidade dos processos da organização. O método de auditoria estabelece os seguintes níveis:

- Inexistente: significa que o processo de gerenciamento não foi implantado.
- Inicial: o processo é realizado sem organização, de modo não planejado.
- Repetível: o processo é repetido de modo intuitivo, isto é, depende mais das pessoas do que de um método estabelecido.
- Definido: o processo é realizado, documentado e comunicado na organização.
- Otimizado: as melhores práticas de mercado e automação são utilizadas para a melhoria contínua dos processos.

O resultado do relatório identifica o grau de evolução dos processos na organização que é avaliada, de modo concreto, com base em informações confiáveis de auditoria e parâmetros de mercado.

A visão geral do modelo CobiT, contendo seus domínios e processos pode ser visualizada naFigura 3.

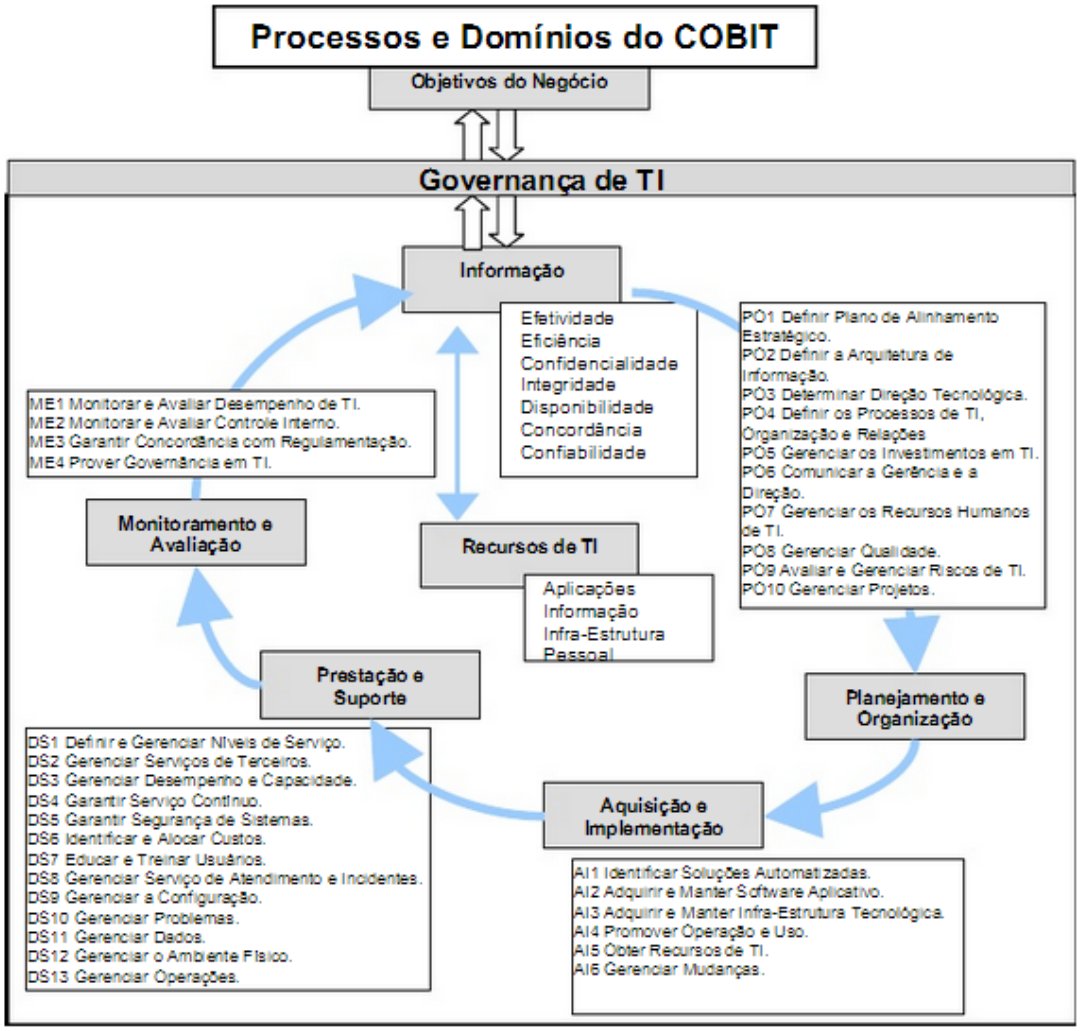

Figura 3 - Processos e domínios do CobiT - COBIT 4.1, (2007)

Dentre os trinta e quatro processos, treze estão aninhados ao domínio *Entregar e Suportar*, sendo que um deles trata especificamente sobre assegurar a continuidade dos serviços, e é neste processo que está alocado o objetivo deste trabalho. Na Figura 3, o processo é identificado pela sigla DS4.

Este processo trata de assegurar que haja um impacto mínimo nos negócios quando houver um problema/interrupção dos serviços de TI. É focado em coligar a capacidade de recuperação em soluções automatizadas, desenvolver, manter e testar os planos de continuidade. O sucesso deste processo é obtido quando houver o desenvolvimento, manutenção e melhoria da contingência de TI, quando houver treinamentos e testes de planos de contingências de TI e quando apresentar armazenamentos em locais remotos de cópias dos dados e dos planos de contingência. É medido por quantidade de horas perdidas por mês pelos

usuários devido à indisponibilidade não planejada dos sistemas ou pela quantidade de processos críticos de negócios dependentes da TI e não contemplados no plano de continuidade da TI.

Para assegurar a continuidade do serviço, o CobiT recomenda desenvolver um modelo para a continuidade da TI a fim de apoiar o gerenciamento da continuidade do negócio em toda a empresa através de um processo consistente, com objetivo de apoiar na determinação das necessidades de capacitação em recuperação da infraestrutura e conduzir o desenvolvimento dos plano de contingencia de TI e recuperação de desastres. Tal modelo deve orientar a estrutura organizacional quanto ao gerenciamento da continuidade, contemplando papéis, tarefas e responsabilidades dos provedores de serviço internos e externos, seus gerenciamentos, clientes e as regras e estruturas para documentar, testar e executar planos de recuperação de desastre e continuidade de TI. O plano também deve contemplar fatores como identificação de recursos críticos, monitoramento e informe de disponibilidade dos mesmos, processamento alternativo e princípios de cópias de segurança e recuperação. COBIT 4.1 (2007)

As medidas de segurança físicas devem ser capazes de efetivamente prevenir, detectar e mitigar riscos relacionados a roubo, temperatura, fogo, fumaça, água, vibração, terrorismo, vandalismo, quedas de energia, produtos químicos ou explosivos.

Controle de acesso é um processo que limita e controla o acesso a recursos de um sistema de computador; um controle lógico ou físico com a finalidade de proteger contra entrada ou uso não autorizados.

Desenvolver planos de continuidade de TI com base na estrutura e projetados para reduzir o impacto de uma grande interrupção de funções e processos de negócio são fundamentais. Os planos devem ser baseados no entendimento do risco de possíveis impactos no negócio, contemplar os requisitos de capacidade de restabelecimento, processamento alternativo e capacidade de recuperação de todos os serviços críticos de TI. Também devem abranger manuais de uso, papéis, responsabilidades, procedimentos, processos de comunicação e abordagens de teste. COBIT 4.1 (2007)

Também é recomendação desenvolver planos de continuidade com base na estrutura e projetados para reduzir o impacto de interrupções de funções e processos fundamentais de negócio. Dar atenção especial aos itens mais críticos no plano de continuidade para garantir a capacidade de restabelecimento e definir prioridades em situações de recuperação. Planejar as ações a serem executadas nos momentos de recuperação e retomada dos serviços, incluindo

ativação de backups, iniciação de processamento alternativo, comunicação para as partes interessadas e os clientes, procedimentos de retorno a produção, entre outros.

Um processo eficaz de continuidade de serviços minimiza a probabilidade e o impacto de uma interrupção de um serviço chave de TI nas funções e processos críticos de negócio.

2.1.2 ABNT NBR ISO/IEC 17799

Segurança da informação é manter a confidencialidade, integridade e disponibilidade da informação. Abrange muito mais do que a segurança da informação de TI, cobre a segurança de toda e qualquer informação da empresa, esteja ela em meios eletrônicos, papel ou até mesmo na memória dos funcionários.

A informação pode existir em diversas formas. Ela pode ser impressa ou escrita em papel, armazenada eletronicamente, transmitida pelo correio ou por meios eletrônicos, apresentada em filmes ou falada em conversas. Seja qual for a forma apresentada ou o meio através do qual a informação é compartilhada ou armazenada, é recomendado que ela seja sempre protegida adequadamente. ABNT NBR ISO/IEC 17799, (2005)

Motivados pela busca de soluções para a segurança da informação, a norma ABNT NBR ISO/IEC 17799 foi elaborada no Comitê Brasileiro de Computadores e Processamento de Dados (ABNT/CB-21), pela Comissão de Estudo de Segurança Física em Instalações de Informática (CE-21:204.01). O Projeto circulou em Consulta Nacional conforme Edital n. 03, de 31.03.2005, com o número de Projeto NBR ISO/IEC 17799. Utilizando-se essa norma, que é um guia de melhores práticas, simplifica-se o trabalho de adoção e implementação de políticas e padrões definidos, bem como da posterior verificação da conformidade dos resultados alcançados. ABNT NBR ISO/IEC 17799, (2005)

Entre outros assuntos, a NBR ISO/IEC 17799 orienta que os equipamentos de TI sejam colocados no local seguro ou protegidos para reduzir os riscos de ameaças e perigos do meio ambiente, bem como as oportunidades de acesso não autorizados. Orienta também que as condições ambientais, como temperatura e umidade, sejam monitoradas para a detecção de condições que possam afetar negativamente os recursos de processamento da informação.

Afirma também que, adicionalmente à notificação de eventos de segurança da informação e fragilidades, o monitoramento de sistemas, alertas e vulnerabilidades seja utilizado para a detecção de incidentes de segurança da informação. Cita como necessário que

um procedimento de notificação formal seja estabelecido para relatar os eventos de segurança da informação, junto com um procedimento de resposta a incidente e escalonamento, estabelecendo a ação a ser tomada ao se receber a notificação de um evento de segurança da informação. Além disso, a NBR ISO/IEC 17799 entende que convém que um ponto de contato seja estabelecido para receber as notificações dos eventos de segurança da informação, e que falhas cometidas pelos usuários ou pelos programas de sistema relacionados a problemas com processamento da informação ou sistemas de comunicação sejam registradas.

A estrutura da norma é composta por 11 seções de controles de segurança da informação, que juntas totalizam 39 categorias principais de segurança e uma seção introdutória que aborda análise/avaliação e o tratamento de riscos.

A seção 9 da norma define as melhores práticas para garantir a segurança física e do ambiente. Dentre elas, destaca-se o foco na segurança dos equipamentos, exigindo a adoção de controles para minimizar o risco de ameaças físicas potenciais, tais como furto, incêndio, explosivos, fumaça, água, poeira, vibração, efeitos químicos, interferência com o suprimento de energia elétrica, interferência com as comunicações, radiação eletromagnética e vandalismo. Destaca-se também a prática que exige que condições ambientais, como temperatura e umidade, sejam monitoradas para a detecção de situações que possam afetar negativamente os recursos de processamento da informação. Estas informações vão diretamente de encontro ao objetivo deste trabalho.

No próximo capítulo, será abordado o Arduino, que faz parte da Tecnologia propriamente dita, e que fará parte da solução proposta por este trabalho, a fim de tratar os problemas evidenciados no capítulo da Gestão da Continuidade dos Serviços.

3 PLATAFORMA ARDUINO

Arduino é uma plataforma eletrônica de hardware livre, projetada com um microcontrolador de placa única, com suporte de entrada e saída embutido, com uma linguagem de programação padrão, com essência em C/C++. Vem ganhando espaço considerável junto a entusiastas, acadêmicos e profissionais do ramo. ARDUINO Site Oficial, (2013).

O Arduino permite a captura de informações do meio através de uma série de sensores que monitoram variações no ambiente, trata essa informação e a converte em grandezas digitais e a partir daí executa um comando pelo desenvolvedor, tal como: controlar luzes, motores ou outras saídas físicas.

Outra plataforma existente é a *Raspberry Pi,* que, segundo Monk (2013), é um computador que executa o sistema operacional *Linux*, e possui portas USB para conexão de teclado e mouse, além de uma saída de vídeo HDMI. É pequeno e comercializado por um preço extremamente acessível, estando à venda no mercado desde fevereiro de 2012.

Apesar de existirem diversas plataformas microcontroladas disponíveis no mercado - como por exemplo a já citada *Raspberry Pi,* e outras como *Netmedia's, Parallax Basic Stamp, Handyboard e Phidgets* – a plataforma Arduino despontou para ser escolhida para este projeto devido a sua popularidade junto a comunidade acadêmica e também por outros fatores citados abaixo:

- Custo x Benefício – a versão mais barata do Arduino pode ser montada manualmente, e, inclusive os módulos montados chegam a custar menos de R$ 50,00(ARDUINO Site Oficial, 2013).
- Hardaware e Software Livres – ambos possuem seus códigos, esquemas e planos publicados na comunidade Open Source, onde qualquer pessoa pode tomar conhecimento da tecnologia, aprimorá-la e continuar seu desenvolvimento mesmo que o criador da tecnologia perca o interesse pela mesma. Ser Open Source é um dos grandes pontos fortes.
- Ambiente de programação: a IDE de desenvolvimento é escrita em Java, multiplataforma, fácil de usar para iniciantes e suficientemente flexível para que usuários avançados possam aproveitá-lo ao máximo.

● Multiplataforma: o software para a programação do Arduino roda em sistemas operacionais Microsoft Windows, Apple Macintosh e Linux.

● Portabilidade: após o desenvolvimento, pelo próprio ambiente de programação, o software é transferido para o Arduino via USB. Após a transferência do software, é possível desligar o Arduino, sem que ele perca o conteúdo.

3.1 Modelos do Arduino

O Arduino original é fabricado pela companhia italiana *Smart Projects*, porém outras empresas mundo afora também possuem algumas marcas comerciais sobre a mesma licença. O site oficial da empresa apresenta 15 modelos de Arduino. Os mais utilizados são:

3.1.1 Arduino UNO

É uma placa com microcontrolador Atmega 328. Possui 14 entradas/saídas digitais, 6 entradas analógicas, um cristal oscilador de 16MHz, conexão USB, uma entrada para fonte, soquetes para ICSP, e um botão de reset. ARDUINO Site Oficial, (2013). A Figura 4 - Arduino UNO ilustra a placa Arduino UNO.

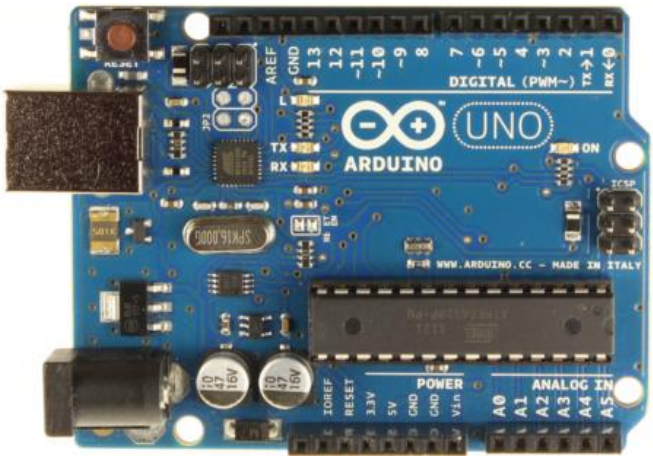

Figura 4 - Arduino UNO

3.1.2Arduino Duemilanove

Arduino Duemilanoveé uma placa de microcontrolador baseada no Atmega328. Possui 14 pinos de entrada/saída digital, dos quais 6 podem ser usados como saídas analógicas PWM, 6 entradas analógicas, um cristal oscilador de 16 MHz, uma conexão USB, uma entrada para alimentação, um cabeçalho ICSP e um botão de reset.

Duemilanove significa 2009 em italiano e o nome foi escolhido pelo ano de lançamento(ARDUINO Site Oficial, 2013). A Figura 5 - Arduino Duemilanove ilustra a placa Arduino Duemilanove.

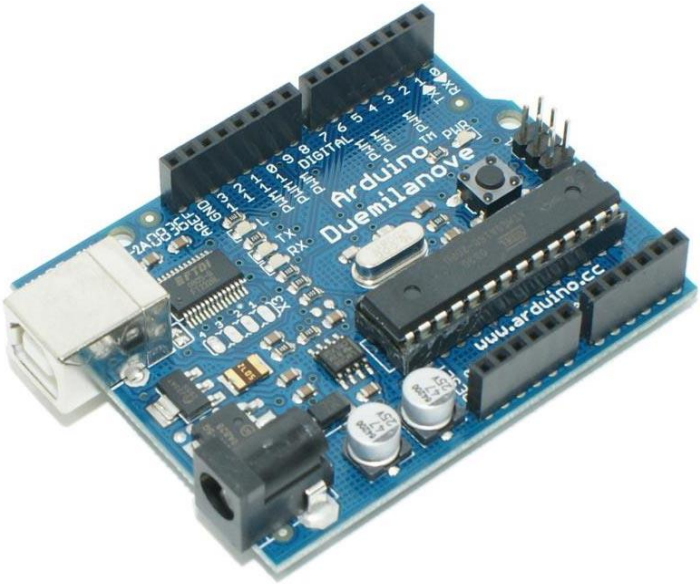

Figura 5 - Arduino Duemilanove

3.1.3Arduino Mini

O Arduino Mini é uma placa de microcontrolador de tamanho reduzido desenvolvida para uso em *protoboards* e em situações em que há limitações de espaço. Baseada no Atmega328, que permite que todos os componentes sejam alocados na parte superior da placa.

Possui 14 pinos de entrada/saída digital, dos quais 6 podem ser usados como saídas analógicas PWM, 8 pinos de entrada analógica, dos quais 4 estão disponíveis na barra de pinos pré soldada. ARDUINO Site Oficial, (2013). A Figura 6 - Arduino Mini ilustra a placa Arduino Mini.

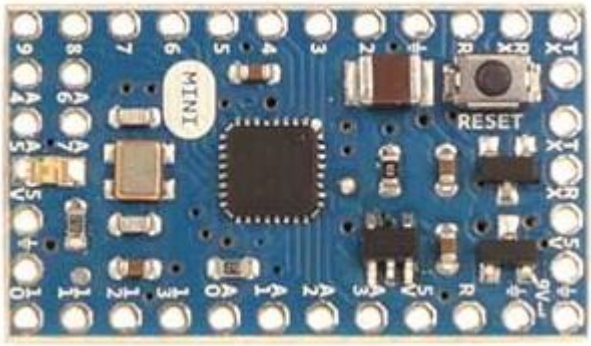

Figura 6 - Arduino Mini

3.1.4Arduino Ethernet

O Arduino Ethernet é uma placa de microcontrolador baseado no Atmega328. Ele tem 14 pinos de entrada/saída digital, dos quais 6 podem ser usados como saídas PWM, 6 entradas analógicas, um cristal oscilador de 16MHz, um conector RJ45, um conector de alimentação um conector ICSP e um botão de reset. ARDUINO Site Oficial, (2013)

O modelo é diferente de outras placas pelo fato de não possuir um chip conversor de USB para serial, mas tem uma interface Ethernet Wiznet que é a mesma interface encontrada no assessório de internet que pode ser acoplado aos demais modelos, chamado de Ethernet Shield.

Uma barra de 6 pinos pode ser conectada a um cabo FTDI ou um conversor USB para serial para fornecer alimentação USB e comunicação com a placa. A Figura 7 - Arduino Ethernet ilustra a placa Arduino Ethernet.

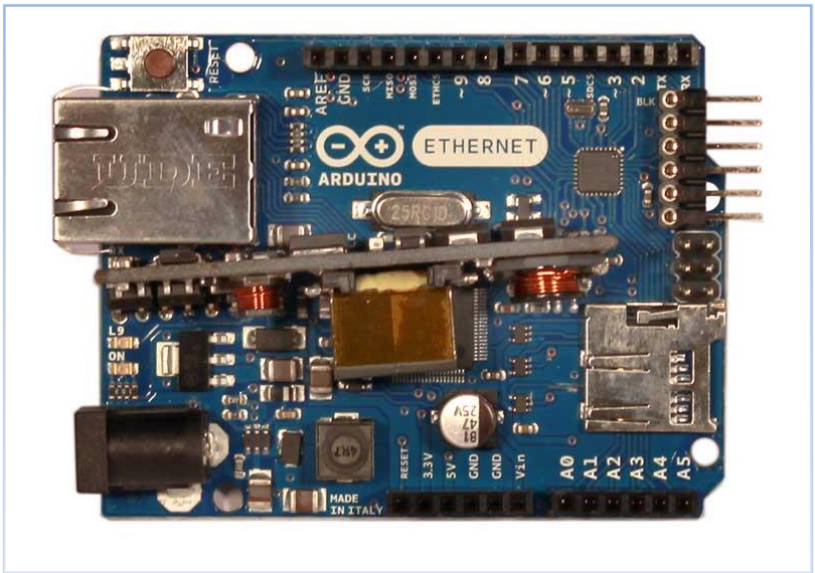

Figura 7 - Arduino Ethernet

3.1.5Arduino Mega 2560

O Arduino Mega 2560 é uma placa de microcontrolador baseada no Atmega2560. Possui 54 pinos de entradas/saídas digitais, 16 entradas analógicas,4 UARTs (portas seriais de hardware), um oscilador de cristal de 16 MHz, uma conexão USB, uma entrada de alimentação, uma conexão ICSP e um botão de reset. ARDUINO Site Oficial, (2013)

Viabiliza a implementação de projetos mais complexos, garantindo eficiência e baixo custo. Será o modelo utilizado para cumprir os objetivos deste trabalho. A Figura 8 - Caracterização do Arduino Mega 2056 ilustra a placa Arduino Mega 2560.

3.1.5.1 Definições

Memória flash: é capaz de preservar os dados armazenados por um longo tempo sem a presença de corrente elétrica.

EEPROM: memória não volátil, pode ser programada e apagada várias vezes, eletricamente.

SRAM: memória de acesso aleatório que mantém os dados armazenados desde que seja mantida sua alimentação elétrica.

Micro controlador: é um "computador-num-chip", contendo um processador, memória e periféricos de entrada/saída. É um microprocessador que pode ser programado para funções específicas, em contraste com outros microprocessadores de propósito gerais (como os utilizados nos PCs). Eles são embarcados no interior de algum outro dispositivo (geralmente um produto comercializado) para que possam controlar as funções ou ações do produto. Outro nome para o micro controlador, portanto, é controlador embutido.

Shields: são placas de circuito impresso com uma função específica, que podem ser acopladas ao Arduino.

3.1.5.2 Especificações

Especificação	*Valor*
Microcontrolador	***Atmmega2056***
Tensão de operação	***5V***
Tensão de entrada recomendada	***7 a 12 V***
Tensão de entrada (limites) 6-20V	***6 a 20 V***
Pinos de entrada e saída (I/O) digitais	***54*** ***(dos quais 14 podem ser saídas PWM)***
Pinos de entradas analógicas	***16***
Corrente DC por pino I/O	***40mA***
Corrente DC para pino de 3,3V	***50mA***
Memória Flash	***256KB (dos quais 8KB são usados para o bootloader)***
SRAM	***8KB***
EEPROM	***4KB***

Velocidade de Clock	***16MHz***

Tabela 1 - Especificações Arduino Mega 2560. ARDUINO Site Oficial, (2013)

3.1.5.3 Alimentação

O Arduino Mega2560 pode ser alimentado pela conexão USB ou com uma fonte externa. A entrada de alimentação é selecionada automaticamente. Alimentação externa (não USB) pode ser tanto de uma fonte como de baterias. A fonte pode ser conectada plugando um conector de 2,1mm, positivo no centro, na entrada de alimentação. Cabos vindos de uma bateria podem serinseridos nos pinos terra (Gnd) e entrada de voltagem (Vin) do conector de energia.

A placa pode operar com alimentação externa entre 6 e 20 volts. No entanto, se menos de 7 volts forem fornecidos o pino de 5V pode fornecer menos de 5 volts e a placa pode ficar instável. Com mais de 12V o regulador de voltagem pode superaquecer e danificar a placa. A faixa recomendável é de 7 a 12 volts. ARDUINO Site Oficial, (2013)

Os pinos de alimentação são os seguintes:

● VIN. Relacionado à entrada de voltagem da placa Arduino quando se está usando alimentação externa (em oposição aos 5 volts fornecidos pela conexão USB ou outra fonte de alimentação regulada). É possível fornecer alimentação através deste pino ou acessá-la se estiver alimentando pelo conector de alimentação.

● 5V. Fornecimento de alimentação regulada para o micro controlador e outros componentes da placa.

● 3V. Uma alimentação de 3,3 volts gerada pelo chip FTDI. A corrente máxima é de 50 mA.

● GND. Pinos terra.

3.1.5.4 Entrada ou Saída

Cada um dos 54 pinos digitais do Mega2560 pode ser usado como entrada ou saída. Eles operam a 5 volts. Cada pino pode fornecer ou receber um máximo de 40 mA e possui um resistor interno de 20-50 KΩ.

3.1.5.5 Pino 13 - LED

No pino 13, existe um LED conectado. Quando o pino está em HIGH, o LED se acende.

3.1.5.6 Memória

O microcontrolador Atmega2560 tem 256 KB de memória flash para armazenamento de código, dos quais 8KB são usados pelo *bootloader*, 8 KB de SRAM e 4 KB de EEPROM.

3.1.5.7 Características Físicas

Na Figura 8 - Caracterização do Arduino Mega 2056está representada a localização dos pinos de Entrada/Saída Digital, as entradas analógicas, as portas de alimentação, entradas de comunicação serial, entre outros.

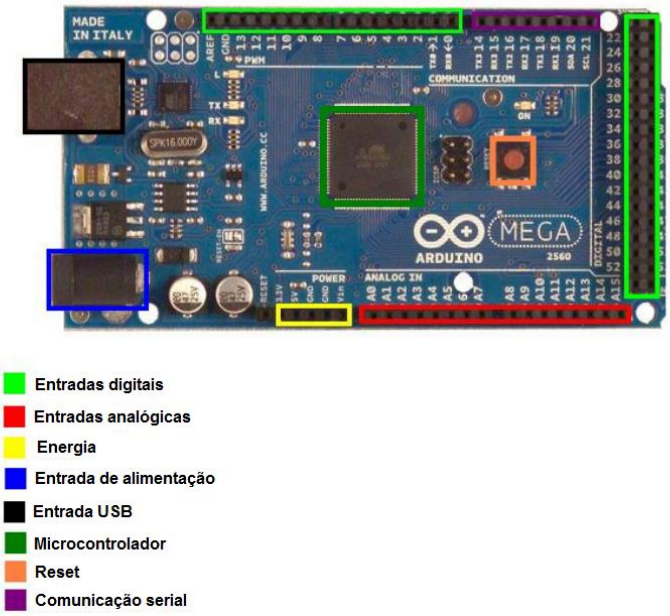

Figura 8 - Caracterização do Arduino Mega 2056

3.2 Shields

Shields são placas que podem ser conectados no Arduino estendendo as suas capacidades. Normalmente elas são fixadas no topo da placa através de uma conexão alimentada por pinos-conectores. São expansões que disponibilizam várias funções específicas, desde a manipulação de motores até sistemas de rede sem fio. Estas placas seguem a mesma filosofia do Arduino: são fáceis de montar e possuem um baixo custo.

São inúmeros os Shields disponíveis para o Arduino, com os objetivos mais diversos, como por exemplo:

- Ethernet: utilizada para conectar o Arduino numa rede cabeada.
- Wireless: utilizada para conectar o Ardunio numa rede sem fio.
- Joystick
- Kit motor
- microSD: usado para permitir leitura e escrita em cartões de memória.
- Módulo de expansão EEPROM
- Relê
- GPS

- GSM / GPRS
- LCD
- MIDI
- MP3 Player
- RFID
- Sensores (acelerômetros, giroscópios, gases, corrente, força e flexão, mecânicos, óticos, ultrassônicos, distância, movimento, temperatura, biométricos, ambientais).

3.3 IDE de desenvolvimento do Arduino

A interface de desenvolvimento do Arduino contém um editor de texto para a escrita de código, uma área de mensagens, um console de texto, uma barra de ferramentas com botões para variadas funções, e uma série de menus. Ele realiza a conexão com o Arduino, permitindo a comunicação com o hardware.

Os projetos escritos utilizando a IDE recebem o nome de sketches. Tais sketches são escritas no editor de texto, e são salvas com a extensão de *.ino. Na área de mensagem, temos retorno de ações como salvar e exportar, além da exibição de erros. O console mostra a saída de texto do Arduino, incluindo mensagens de erro completas e outras informações. O canto inferior direito mostra a *board* atual e a portal serial. Os botões da barra de ferramentas permitem a execução de *upload* de programas, criar, abrir e salvar sketches, e abrir o monitor serial. A Figura 9 - IDE de Desenvolvimento ilustra o IDE do Arduino.

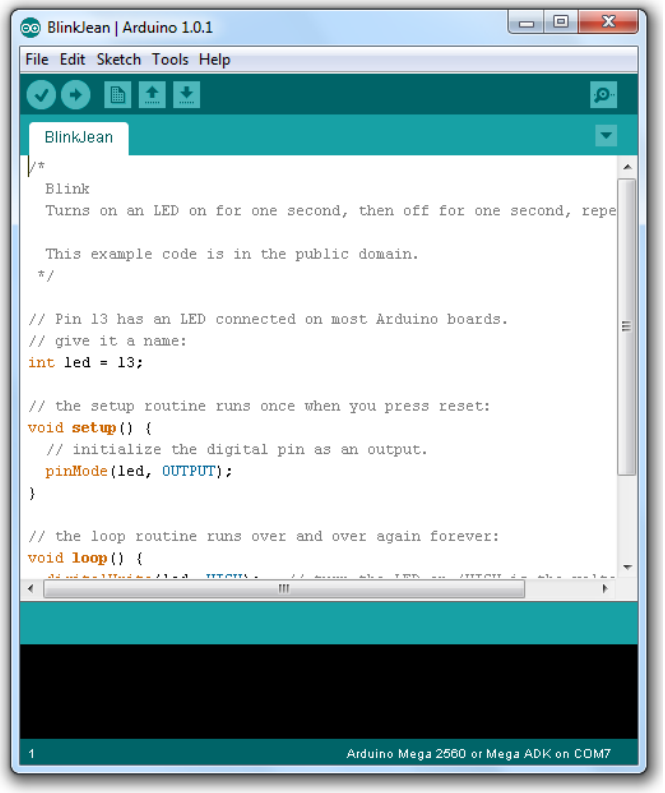

Figura 9 - IDE de Desenvolvimento

3.4 Sensor de temperatura e umidade DHT11

O sensor DHT11 funciona com uma corrente de 5v, padrão do Arduino, a sua faixa de medição para temperatura é entre 0 e 50 graus célsius e a sua taxa para a medição de umidade pode variar entre 20 e 80%. Utiliza três pinos, sendo um de corrente (Vcc), um fio terra (GND) e um de dados (Data). A Figura 10 - DHT11 Sensor de temperatura e umidade. ilustra o sensor de temperatura e umidade.

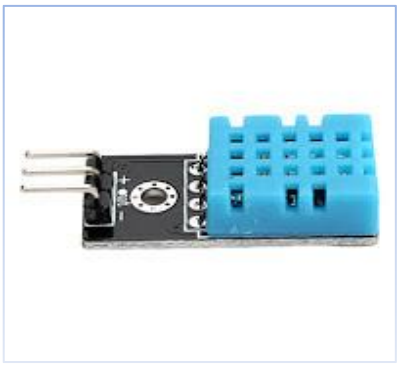

Figura 10 - DHT11 Sensor de temperatura e umidade.

3.5 Sensor de corrente 30A ACS712

O sensor 30A ACS712 é alimentando por uma corrente de 5v, padrão do Arduino, pode fazer leitura de voltagens como 110 e 220v. É de alta confiabilidade, pois é construído com fios de cobre de alta resistência e é capaz de suportar descargas elétricas até 5 vezes a mais do que se esta dando de entrada por padrão, Datasheet 30A ACS712. Também é capaz de identificar oscilações de energia, como no caso de um nobreak estar sobrecarregado ou apresentando falhas de alimentação. A Figura 11 - 30A ACS712 Sensor de corrente elétrica.ilustra o sensor de corrente elétrica.

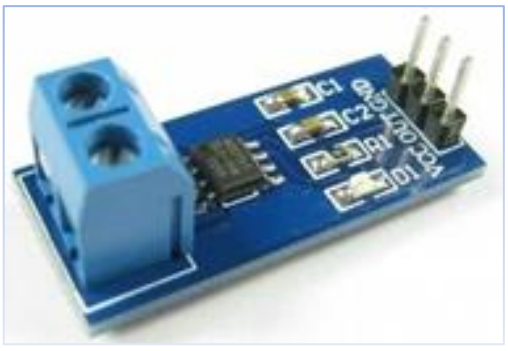

Figura 11 - 30A ACS712 Sensor de corrente elétrica.

O sensor 30A ACS712 utiliza três pinos, sendo um de corrente (Vcc), um fio terra (GND) e um de dados (OUT). Existem outras duas entradas para colocar o fio terra e de

corrente elétrica, onde serão fixados os cabos de saída da tomada da sala de servidores e do nobreak.

3.6 Sensor de fumaça e gases inflamáveis MQ-2

O sensor MQ-2 é alimentado por um corrente de 5v. É capaz se fazer leituras das menores oscilações que ocorrem no ambiente da sala de servidores. A Figura 12 - MQ-2 Sensor de fumaça e gases inflamáveis. ilustra o sensor de fumaça e gases inflamáveis.

Figura 12 - MQ-2 Sensor de fumaça e gases inflamáveis.

O sensor MQ-2 possui quatro pinos, sendo eles Alimentação (VCC), fio terra (GND), saída de dados analógica (AOUT) e saída de dados digital (DOUT). Para este projeto será necessário apenas os pinos VCC, GND e AOUT.

3.7 Modulo RFID Mfrc522

O modulo RFID Mfrc522 é um identificador de rádio frequência que funciona da seguinte maneira: o responsável pela sala de servidores tem um cartão ou um chaveiro (TAG) que o identifica. Ao passar com este cartão pela frente do modulo RFID Mfrc522, o sistema o identificará através de rádio frequência. Ao ser identificado o Arduino enviará sinal para um

relé para que a porta seja destravada, liberando assim o acesso à sala de servidores. A Figura 13 - Modulo RFID Mfrc522 identificador de rádio frequência. ilustra o módulo identificador de rádio frequência.

Figura 13 - Modulo RFID Mfrc522 identificador de rádio frequência.

O modulo RFID Mfrc522 trabalha em 3.3v, sua frequência de identificação de 13,56 Mhz, os cartões ou chaveiros (TAG) que podem ser utilizados são Mifare1 S50, S70 Mifare1, Mifare UltraLight, Mifare Pro, Mifare Desfire. É possível utilizar quantos cartões (ou chaveiros de identificação) forem necessários sendo que cada um deve ter uma chave de segurança diferente.

3.8 Luz de LED RGB

A luz de LED RGB pode mesclar diversas cores a partir das cores primarias, vermelho, verde e azul. Possui quatro pinos sendo eles Vermelho, fio terra (GND), verde e por ultimo azul, cada um dos pinos de cores podem ser acessos de forma independente o que permite criar outras cores além das cores primarias. A Figura 14 - Luz de LED RGB ilustra a luz de LED RGB.

Figura 14 - Luz de LED RGB

3.9 Relé

Relé é capaz de abrir ou fechar a corrente de energia para ligar ou apagar a luz externa do servidor. Possui três pinos para ligar com o Arduino sendo um para alimentação, outro para habilitar e desabilitar a passagem de energia e um último para passagem de energia. A Figura 15 - Relé ilustra o Relé.

Figura 15 - Relé

3.10 Shield Modulo V3.0

O *Shield*Module V3.0 possui suporte para um cartão SIM, saída e entrada de áudio, sua alimentação pode ser entre 6v e 12v, quad-band 850/900/1800/1900 MHz, um baixo

consumo de energia e leds indicadores de status. A Figura 16 - Module V3.0 componente para envio de SMS ilustra o componente para envio de SMS.

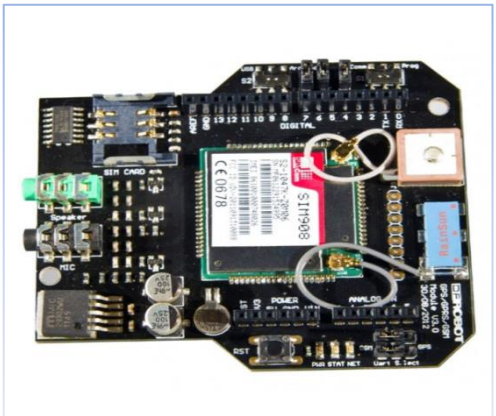

Figura 16 - Module V3.0 componente para envio de SMS

Este componente é de alta capacidade e de ampla usabilidade. Neste projeto será necessário apenas um module V3.0(ROBOCORE, 2013).

3.11 Ethernet Shiled R3

O Componente Ethernet Shield R3 é capaz de conectar em internet de banda larga como 10/100 mbps, cabo de conexão padrão RJ-45, alimentação de 5v. Este componente é acoplado acima do Arduino usando seus respectivos pinos de leitura a gravação bem como os de alimentação.Figura 17 - Ethernet Shield R3 componente para envio conexão com a Internet. ilustra o componente para conexão do Arduino a rede cabeada.

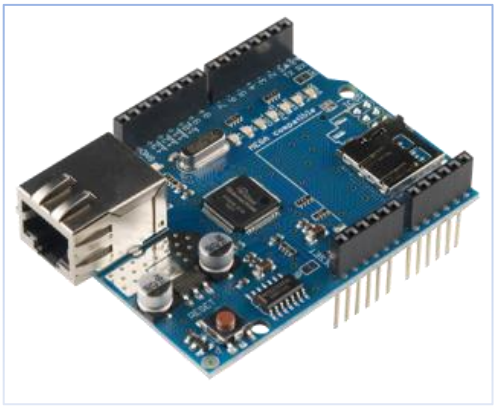

Figura 17 - Ethernet Shield R3 componente para envio conexão com a Internet.

Este componente também possui um slot para um cartão Micro-SD para gravar arquivos e demais informações podendo ser acessado e enviado pelo Arduino. Este cartão Micro-SD será de grande importância para o item 4.3.1 deste projeto, pois permitirá armazenar dados que posteriormente serão enviados.

3.12 Valores

ATabela 2 - Valores Arduino e Shields. destaca o valor aproximado de mercado para as diferentes versões do Arduino e de seus Shields. Os valores são aproximados, pesquisados em sites especializados na internet, como por exemplo, nos sites: http://www.webtronico.com/, http://www.dx.com e http://robocore.com.br/.

Quantidade	Descrição	Valor
1 un.	Arduino Mega 2560	R$ 90,00
1 un.	Sensor DHT11	R$ 15,00
2 un.	Sensor 30A ACS712	R$ 40,00
1 un.	Sensor MQ-2	R$ 20,00
1 un.	Modulo RFID Mfrc522	R$ 60,00
4 un.	LED RGB	R$ 20,00
1 un.	Relé	R$ 15,00
1 un.	Shield Modulo V3.0	R$ 500,00

1 un.	Ethernet Shiled R3	R$ 150,00
Total:		**R$ 910,00**

Tabela 2 - Valores Arduino e Shields.

4 TRABALHOS RELACIONADOS

A pesquisa, ora apresentada, mostra-se importante para a área de automação, uma vez que tem como objetivo desenvolver um protótipo de baixo custo, que utilize a tecnologia ARDUINO, para monitoramento e registro de recursos de TI em sala de servidores. Ainda, essa importância resta evidenciada, quando se verifica que não há um conhecimento e muito menos aproveitamento do potencial dessa tecnologia emergente no mundo empresarial.

Assim o presente capítulo apresenta trabalhos relacionados, que utilizam a tecnologia ARDUINO, abordando a existência de outras produções técnicas que obtiveram êxito na criação dessa plataforma de trabalho.

Inicialmente, destaca-se o trabalho de Iserhardt (2010), onde é apresentada a solução para a identificação e notificação para o controle de temperatura de um *datacenter*. Tendo como saídas de dados avisos sonoros e visuais, o projeto atende somente a parte de superaquecimento do ambiente da sala de servidores para evitar o mau funcionamento dos equipamentos.

Já no trabalho abordado por Oliveira (2012), é realizado um controle de automação residencial, visando conceitos de internet das coisas, controlando todo o ambiente através de uma interface web, onde o usuário poderia desligar uma luz, ligar a televisão, controlar micro-ondas, entre outras funções, visando facilitar e agilizar o dia-a-dia de seus moradores, utilizando uma tecnologia de baixo custo e fácil utilização.

Na mesma vertente, é a pesquisa de Pinto (2011), que possui como foco a formação de professores na área da robótica educacional, sustentado sob dois pilares: um pedagógico, com a aplicação de uma arquitetura interativa, apoiada nas tecnologias de informação e comunicação (TIC) e outro tecnológico, com a proposta de utilização de tecnologias livres de hardware, como o projeto Arduino, objetivando o acesso de instituições públicas de ensino a modernas plataformas de programação, seja pelo fator custo, seja pela facilidade de programação por não especialistas em informática e eletrônica, como os professores da educação básica.

Brugnari & Maestrelli (2010), por sua vez, apresentam uma pesquisa que possui como base uma interface web para os usuários poderem acionar equipamentos que

estejam ligados à rede elétrica, a fim de que seja possível realizar as suas tarefas sem estarem no ambiente em questão, ou seja, são tratados e conceitos como internet das coisas, que utiliza Arduino como base de hardware para o desenvolvimento do projeto.

Focando na mesma temática, Silva et. al. (2012) apresenta o desenvolvimento de um veículo terrestre tele operado, para explorar os conceitos presentes nas áreas de Sistemas Embarcados e Robótica Móvel. Esse trabalho foi desenvolvido com base na Plataforma de Desenvolvimento, tendo como um fator positivo, o fato de que o Arduino, que funciona como mediador entre a teoria e a prática, tem como filosofia básica o princípio de facilitar à eletrônica e torná-la mais atraente aos que querem ingressar na área computacional embarcada.

Pinto F. D. (2010) nos traz que a automação de habitações – domótica – é um assunto que suscita o interesse da população em geral, visto que a existência de sistemas para realizar tarefas rotineiras, tornar a habitação mais segura e ao mesmo tempo permitir poupar recursos como energia eléctrica, alicia o comum dos consumidores. Um sistema domótico agrega um conjunto de tecnologias e de mecanismos que funcionam em conjunto e como tal, necessitam ser facilmente integráveis para melhor servirem o utilizador. As soluções disponíveis atualmente possuem diversas restrições das quais se destacam a sua incapacidade de interatuarem com outras tecnologias, dificuldade sua evolução e adaptação a novas necessidades dos utilizadores, além de possuírem um elevado custo.

Neto (2011), apresenta a especificação e implementação de um protótipo para controle de automação residencial à distância utilizando um iPad e um microcontrolador Arduino. Esse conjunto de dispositivos tem como objetivo automatizar a iluminação de uma residência, provendo facilidade e dinamismo para usuários deficientes ou não. De forma inovadora o tablet iPad poderá controlar toda a iluminação de uma residência através de simples toques em tela e comunicação Wi-Fi.

5 LEVANTAMENTO DE REQUISITOS E TECNOLOGIA UTILIZADA

Este capítulo aborda os requisitos da aplicação juntamente com as tecnologias utilizadas para o desenvolvimento da aplicação.

5.1 Variáveis do ambiente

Em salas de servidores precisamos medir variáveis do ambiente como, temperatura, umidade, energia elétrica, sensores de fumaça e identificação de acessos. Também é dever do projeto fazer com que os responsáveis pela sala de servidores possam ser notificados através de SMS, e-mail ou sinais luminosos, sobre falhas ou alterações nas variáveis de ambiente, para que possa ser tomada alguma medida a fim de diminuir o risco de danos aos equipamentos e indisponibilidade dos serviços.

Nas próximas paginas será abordado de forma detalhada cada passo que o projeto pretende abranger.

5.1.1 Temperatura e Umidade

É importante monitorar e controlar a temperatura do ambiente para evitar superaquecimento dos servidores e também para que possam ser tomadas iniciativas e realizadas ações para não danificar os equipamentos, evitando possíveis danos e falhas nos serviços.

A identificação de umidade relativa do ar em uma sala de servidores é importante, pois com ela é possível identificar se o sistema de arrefecimento e refrigeração do ambiente está operando conforme o esperado, sendo assim possível identificar falhas ou correções para um melhor funcionamento dos equipamentos. Este tipo de identificação evita que haja uma perda

de dados e possíveis falhas nos serviços da empresa, bem como a perda de investimento em equipamentos e estrutura.

A temperatura ideal para funcionamento da sala de servidores gira em torno de 12 a 24 graus Celsius, mas devem ser avaliadas com cuidado, pois valores muito próximos dos limites podem indicar que o sistema de arrefecimento esteja apresentando defeitos. A umidade relativa do ar pode variar entre 20% e 80% sem que haja problemas, porém a mesma regra da temperatura pode ser aplicada nesta situação, sendo que valores muito próximos dos limites podem estar sendo gerados por falhas no sistema de arrefecimento. Veras & Tozer (2012)

Para monitorar as grandezas apresentadas acima será utilizado o sensor DHT11, que mede a temperatura e a umidade relativa do ar. No projeto, apenas um sensor será utilizado.

5.1.2 Energia elétrica

Outro fator de grande importância é a energia elétrica. Quando houver uma falha na energia elétrica da sala de servidores pode-se significar problemas como indisponibilidade de serviços, perda de dados, queima de aparelhos e componentes elétricos.

Atualmente grande parte das empresas opta por adquirir um *nobreak* para solucionar este problema, porém, o mesmo também deve ser monitorado, pois também esta sujeito a falhas e sua duração de energia pode variar muito dependendo do tempo de uso e da quantidade de aparelhos que estão utilizando seus recursos.

Para evitar este tipo problema, serão utilizados dois sensores 30A ACS712. Um para fazer a leitura de energia elétrica da sala de servidores e outro para fazer a leitura de saída de energia do *nobreak*, para verificar se ele está danificado, sobrecarregado ou sem energia.

5.1.3 Sensor de Fumaça e gases inflamáveis

Identificação de fumaça e gases inflamáveis é uma questão de alta relevância, e que também devem ser monitorados.

A fumaça e gases inflamáveis podem ter inúmeras origens, como vazamento de gás no prédio, falha no sistema de arrefecimento dos servidores, problemas no sistema de climatização da sala de servidores, queima ou má funcionamento de componentes elétricos.

Os problemas citados anteriormente podem acarretar danos em todos os equipamentos da sala de servidores, bem como a perda de dados, podendo danificar não somente uma máquina, mas toda a rede da empresa, ocasionando prejuízo nos investimentos e até mesmo uma indisponibilidade de serviços.

Uma solução para evitar este tipo de problema é fazer a identificação de nível de gases inflamáveis e fumaça na sala de servidores.Com isso o responsável tem tempo hábil para tomar alguma iniciativa para sanar o problema ou evitar danos aos equipamentos.

Para fazer a identificação de gases inflamáveis e fumaça será usado um sensor MQ-2, no projeto inicialmente será necessário apenas um sensor de fumaça e de gases inflamáveis, pois este sensor identifica a oscilação de todo o ar do ambiente da sala de servidores.

5.2Notificações e Alertas

Todas as informações que serão geradas a partir dos sensores de temperatura, umidade, energia, fumaça e de controle de acessos deverão ser notificados ao administrador. Tal notificação pode ser feita através de avisos sonoros, luzes de LED acesas com diversas cores para melhor identificação, envio de SMS para o celular de responsável pelo servidor e também através de envio de e-mail.

Para ter um registro de ocorrências será armazenado um log de alterações nas variáveis da sala de servidores para possíveis consultas, também será usado este mesmo log para o controle de contingência que será abordado no item 5.3. Estes logs serão armazenados em um cartão Micro-SD através do componente Ethernet Shield R3 que será detalhado no item 5.2.3.

5.2.1 Luzes de status

Para fazer uma rápida identificação caso exista algum problema com o ambiente dos servidores, será efetuado o uso de luzes. Luz vermelha para estado crítico, amarelo para informar que algum ponto está apresentando avarias, porém não está em estado crítico, e verde para quando o ambiente estiver dentro dos padrões de funcionamento da sala de servidores.

Para fazer a identificação de status das variáveis do ambiente da sala de servidores, serão utilizadas cinco luzes RGB. Uma para temperatura, uma para umidade, uma para energia elétrica, uma para sensor de fumaça e uma ultima de para gases inflamáveis.

Estas cinco luzes irão, quando o ambiente do servidor estiver funcionando corretamente se manter na cor verde. Porém quando houver uma oscilação na temperatura para um valor que não é esperado, a luz que identifica o status da temperatura irá trocar de cor para vermelho ou se for uma mudança que não compromete o funcionamento do servidor ficará em amarela.

Será necessária uma luz na parte externa da sala de servidores, para auxiliar no alerta de eventuais problemas. Para esta luz de identificação externa ser ligada e desligada será necessário fazer o uso de um componente de relé, que ao ser acionado terá a função de bloquear ou deixar passar a energia.

Quando as variáveis de ambiente sofrerem alteração, a luz externa irá acender e quando elas voltarem aos valores padrões a luz externa se apagará.

5.2.2 Envio de SMS

Em casos de problemas, os responsáveis pelo servidor devem ser informados das alterações nas variáveis do ambiente da sala de servidores, podendo assim tomar uma iniciativa para poder contornar a questão o quanto antes, para evitar possíveis perdas ou indisponibilidade de serviços.

Uma das alternativa de se fazer a notificação da alteração nas variáveis doambiente poderá ser efetuada através do envio de SMS, que tem uma grande abrangência e é bastante eficaz, pois a partir do momento que é enviado em poucos segundos o destinatário recebe a informação. Sendo assim, o administrador do servidor toma conhecimento do problema em poucos segundos ou no máximo em poucos minutos.

Para este feito, deve-se utilizar um componente de envio de SMS, que tenha embarcado a tecnologia de GSM, que possibilita o envio e recebimento de SMS além de realizar e receber ligações.

O *Shield* utilizado será Module V3.0,sendo acionado toda a vez que houver uma alteração no ambiente. Ao acionar este componente o SMS será enviado para o responsável com o número já pré-programado no sistema. Este SMS irá conter informações sobre qual a alteração do ambiente foi identificada e o grau de relevância, bem como o valor que estava anteriormente e o novo valor.

5.2.3 Envio de E-mail

Outra forma de distribuição de logs e também de notificações de alteração em variáveis de ambiente acontece através do envio de e-mails, que são de comum usabilidade.

O envio de e-mail será realizado a partir de uma função do PHP mail que possui parâmetros de destinatários, assunto, mensagem e cabeçalhos de envio. Essa chamada será feita através do componente Ethernet Shield R3.

Este componente é de grande importância para o projeto, pois ele será capaz de fazer as notificações a armazenar dados para serem enviados futuramente, como medida de contingência.

5.3 Plano de contingência

Segundo PMBOK® Guide 2004 Edition, determinados riscos não podem ser gerenciados de forma proativa, o que sugere que a equipe do projeto deva criar um plano de contingência.

O plano de contingência deve ser aplicado no projeto, pois falhas podem ocorrer e o sistema deve possuir um meio de ser controlado para que alguma iniciativa seja tomada em caso de falhas.

5.3.1 Armazenar Registros

Em caso de falta de sinal para envio de mensagens via SMS e também ausência de internet o sistema ficaria impossibilitado de enviar mensagens informando o status do ambiente da sala de servidores.

Para contornar este problema será usada a entrada de memória do componente Ethernet Shield R3 para armazenamento de dados, que já foi citado anteriormente no item 5.2.3. O ideal é que nesta memória sejam armazenados todos os registros de ocorrência da sala de servidores, para que seja feito o envio das informações para os responsáveis pela sala de servidor. Assim que houver uma comunicação por sinal de GSM, deverá ser enviado um SMS ou ter acesso à internet para envio de e-mail.

5.3.2 Ausência de energia

Em caso de falta de eletricidade na sala de servidores, o sistema não pode deixar de operar, inclusive ele deve informar aos responsáveis que houve uma falha de energia elétrica no prédio.

O sistema será alimentado por três baterias de 9 volts recarregáveis que estarão entre a saída de energia da sala do servidores e o Arduino, com os devidos transformadores de voltagens.

Com esta implantação, ao faltar energia no prédio o sistema do Arduino não será afetado, pois as três baterias 9 volts irão manter o Arduino ligado por tempo suficiente para ser feito os avisos e notificações necessárias via SMS ou e-mail, bem como armazenar as informações para envio futuros aos responsáveis pela sala de servidores.

5.4 Fluxograma dos processos

O fluxograma apresentado a seguir servirá para demonstrar de forma simples e direta os processos que serão realizados pelo projeto, tanto no seu funcionamento como no plano de contingência.

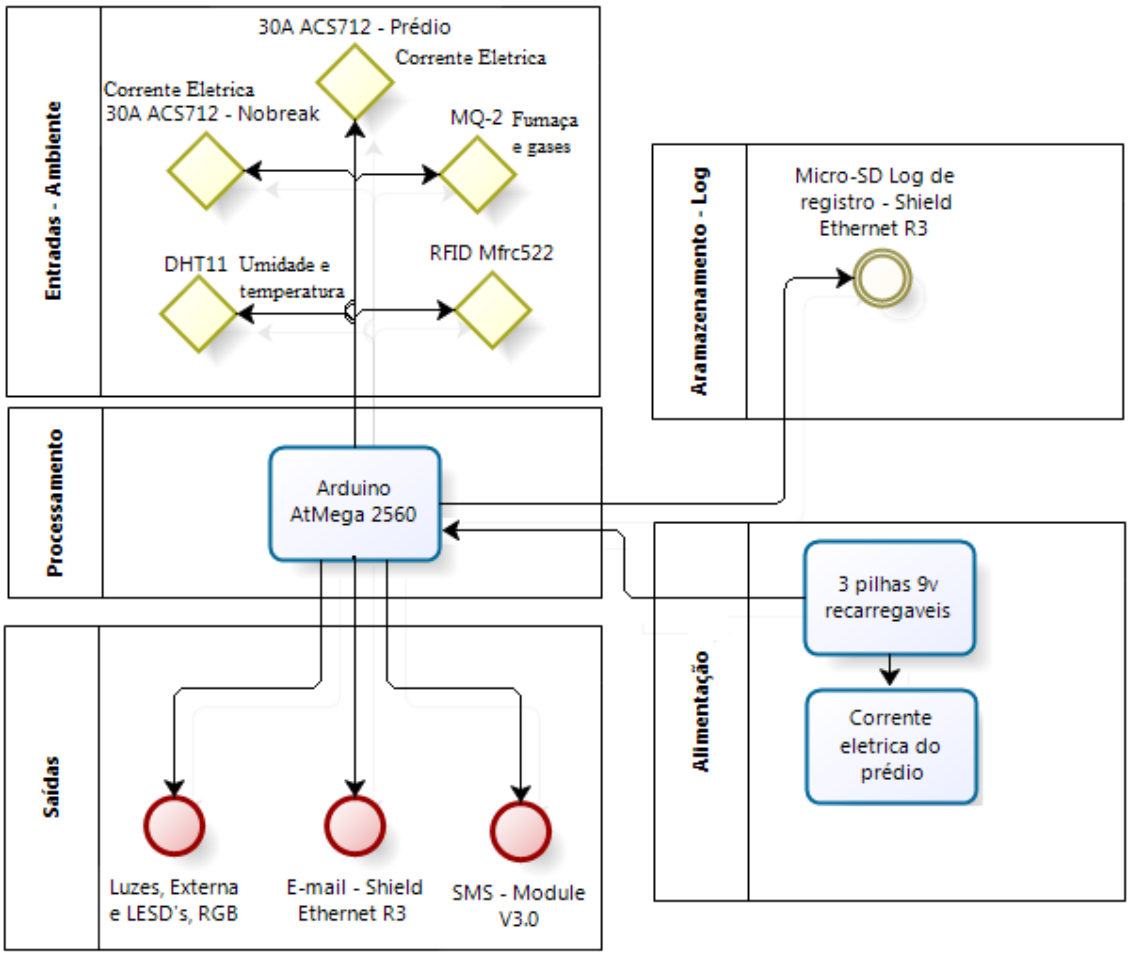

Figura 18 - Fluxograma dos processos.

5.5Tecnologias

Por se tratar de um sistema adaptativo e suprir diversas necessidades de controle do ambiente da sala de servidores, todo o sistema deverá possuir a possibilidade de realizar configurações especificas para cada situação de uso.

Pensando nisto, será criado uma interface onde o usuário possa fazer configuração de quais sensores deseja utilizar, configurar o envio de e-mail e SMS para os responsáveis por cada sensor da sala de servidores, podendo informar valores mínimos e máximos de cada leitura dos sensores.

A aplicação também deverá possuir um registro de log de todas as leituras da sala de servidores, podendo apresentar a leitura de cada sensor em forma de gráfico de linhas. Esta tela de registro de logs irá auxiliar a tomada de decisão para possíveis melhorias ou correções nos equipamentos da sala de servidores.

A aplicação será desenvolvida para ser acessada via internet, através de computadores convencionais, notebook, *smarthphones* e *tablets*, não existe restrição de arquitetura e nem de sistema operacional dos dispositivos que irão acessa-la, tornando assim o projeto mais dinâmico e acessível. Para desenvolver a aplicação na internet será utilizado HTML5, PHP 5, MySQL 5.5, CSS e Javascript.

5.6 HTML 5

Segundo Schroeder (2012) HTML 5 é a nova versão da principal linguagem para construção da Internet: o HTML (Linguagem de Marcação de Hipertexto). Sendo que esta linguagem surgiu principalmente por uma grande pressão do mercado (tanto desenvolvedores como fabricantes de navegadores), que descontentes com o rumo que a Web tomava resolveram fazer as suas novas definições e, acima de tudo padronizá-las. Esta iniciativa teve uma boa recepção e acabou se tornando o que hoje é conhecido como HTML5. Afirma também, que as novas tecnologias do HTML5 estão disponíveis nos mais diversos navegadores do mercado.

Esta nova versão traz consigo importantes mudanças quanto ao papel do HTML no mundo da Web, através de novas funcionalidades como semântica e acessibilidade. Com novos recursos, antes só possíveis por meio de outras tecnologias. Braun (2010)

5.7 PHP

O PHP é uma linguagem de programação de uso geral criada especialmente para trabalhar em ambientes Web. Diferentemente de outras linguagens como o Javascript, o PHP é executado no servidor. Com isso seu código fonte nunca é revelado ao usuário final. E, diferentemente de linguagens como C e Pascal, o PHP não é uma linguagem compilada e sim interpretada, dispensando declarações rígidas de variáveis e permitindo uma mistura de códigos em HTML 5 (HyperText Markup Language) e PHP.

Criado em 1994, a partir do sucesso do Personal Home Page Tools de RasmusLerdof, o PHP chamou atenção de Zeev Suraski e, em 1997, se tornou um pré-processado e hipertexto chegando à versão 3. Em 2000, o PHP atingiu a versão 4 trazendo consigotecnologias da Zend Technologies, que permitiram funções mais complexas e maior interação com o usuário. Oliveiro (2001).

O PHP já se mostrou superior em quesitos como simplicidade de conexão a bancos dedados, desempenho e gerenciamento de memória, além de ser distribuído sob licença GPL e de rodar em inúmeras plataformas. Essa licença torna o PHP *open source* e traz novas vantagens como o grande número de colaboradores distribuídos pelo mundo e o custo extremamente baixo de implementação, sendo apenas necessário investir no hardware e utilizar um conjunto completo de ferramentas gratuitas e *open source*.

Conforme PHP (2013), em sua versão mais recente, o PHP5, foi introduzido um novo modelo de objetos e, com isso, foram implementadas muitas funções e características firmando o PHP como linguagem orientada a objetos, e o colocando a novos níveis de concorrência com linguagens mais robustas, como o Java.

5.8MYSQL

Trata-se de um banco de dados de código aberto sendo atualmente, um dos mais populares de sua categoria. Sua arquitetura permite que seja extremamente rápido e simples de usar e sua distribuição gratuita é um grande atrativo para programadores e empresários que desejam publicar sites na internet.

Com uma estrutura robusta, o MySQL é totalmente capaz de responder e atender soluções web de pequeno a grande porte, com a vantagem de utilizar menos recursos do hardware, em comparação a servidores comercias. Mesmo com essas simplificações, o banco implementa a linguagem SQL (*Structured Query Language*) amplamente utilizada na realização de buscas em banco de dados. Oliveiro (2001).

6 . DESENVOLVIMENTO DA APLICAÇÃO

Este capítulo aborda o desenvolvimento da aplicação.

6.1Casos de Uso

Casos de uso são uma forma de narrar como o sistema irá se comportar para realizar uma função interna, bem como modelar a fluxo de como o processo deverá ser realizado para facilitar o entendimento tanto para os desenvolvedores como para os usuários finais.

6.1.1Cadastrar usuários

Atores: Administradores.
Pré-condições: Possuir os dados do novo usuário a ser cadastrado.
Pós-condições: O novo usuário poder acessar o sistema.
Requisitos: Estar logado no sistema como administrador.
Fluxo:
1. Acessar menu de gerência de usuários.
2. Informar o nome, login e senha.
3. Clicar em salvar.

6.1.2Alterar usuários

Atores: Administradores.

Pré-condições: Possuir os novos dados dos usuários a serem alterados.

Pós-condições: As informações dos usuários devem ser atualizadas.

Requisitos: Estar logado no sistema como administrador.

Fluxo:

1. Acessar menu de gerencia de usuários.
2. Localizar o usuário que deve ser alterado.
3. Clicar em alterar.
4. Serão habilitados os campos de nome e senha para alteração.
5. Clicar em salvar.

6.1.3Excluir usuários

Atores: Administradores.

Pré-condições: Os usuários estarem cadastrados.

Pós-condições: O usuário não poderá mais acessar o sistema.

Requisitos: Estar logado no sistema como administrador.

Fluxo:

1. Acessar menu de gerencia de usuários.
2. Localizar o usuário que deve ser excluido.
3. Clicar em excluir.
4. Exibir mensagem, questionando se deseja realmente excluir o usuário.

6.1.4Configurar sensores

Atores: Administradores.

Pré-condições: Os sensores estarem configurados no Arduino.

Pós-condições: Estas configurações serão importadas pelo Arduino.

Requisitos: Estar logado no sistema como administrador.

Fluxo:

1. Acessar menu de configurações.

2. Localizar o sensor a ser alterado.

3. Os dados que poderão ser alterados são: Tempo de leitura, pino, valor máximo ativo, valor máximo, valor mínimo ativo, valor mínimo e se o sensor esta ativo ou não.

4. Clicar em salvar.

6.1.5Adicionar contatos dos sensores

Atores: Administradores.

Pré-condições: Possuir os dados de contato do sensor.

Pós-condições: Estas configurações serão importadas pelo Arduino.

Requisitos: Estar logado no sistema como administrador.

Fluxo:

1. Acessar menu de configurações.
2. Localizar o sensor que será adicionado um novo contato.
3. Informar o e-mail e telefone do contato
4. Clicar em salvar.

6.1.6Alterar contatos dos sensores

Atores: Administradores.

Pré-condições: Contato estar cadastrado e possuir os novos dados do contato.

Pós-condições: Estas configurações serão importadas pelo Arduino.

Requisitos: Estar logado no sistema como administrador.

Fluxo:

1. Acessar menu de configurações.
2. Localizar o sensor e o contato a ser alterado.
3. Clicar em alterar.

4. Serão habilitados os campos de e-mail e telefone do contato.

5. Clicar em salvar.

6.1.7Excluir contatos dos sensores

Atores: Administradores.

Pré-condições: Os contatos estarem cadastrados.

Pós-condições: Estas configurações serão importadas pelo Arduino.

Requisitos: Estar logado no sistema como administrador.

Fluxo:

1. Acessar menu de configurações.
2. Localizar o contato que deve ser excluído.
3. Clicar em excluir.
4. Exibir mensagem, questionando se deseja realmente excluir o contato.

6.1.8Consultar logs

Atores: Todos os usuários.

Pré-condições: Terem sido feitas as leituras dos sensores no arduino.

Pós-condições: Montar gráfico de leituras.

Requisitos: Estar logado no sistema.

Fluxo:

1. Acessar menu de registro de LOG.
2. Localizar os registros, podendo filtrar por data inicio e data de fim, e por sensor.
3. Clicar em buscar.
4. Serão listados todos os registros de logs de acordo com os filtros.
5. Conforme os dados que foram mostrados, será mostrado um gráfico com base nos dados de registro.

6.2Interface com o usuário

A interface com o usuário será apresentada de forma simples, porém com grande funcionalidade. O menu principal contará com três botões para acessar as telas de gerência de usuários, configuração de sensores e registros de log.

Figura 19 - Menu principal da aplicação.

Este menu irá mostrar de forma rápida e clara ao usuário as funções disponíveis após realizado o *login* na aplicação da internet.

6.2.1Gerência de usuários

A tela de gerência de usuários estará disponível somente para o usuário administrador. Partindo desta tela, o usuário poderá lançar novos usuários, fazer alterações nos usuários atuais e até mesmo excluir os usuários que já foram cadastrados.

Figura 20 - Tela para gerenciar os usuários.

6.2.2Configuração de sensores

Esta tela será acessível apenas para o usuário administrador. É através dela que o usuário poderá informar quais os sensores que serão utilizados na aplicação; o tempo de leitura de cada sensor; qual o pino de leitura do Arduino; qual o pino de saída do Arduino; se possuir controle de valor máximo, qual é o valor máximo; se tem controle de valor mínimo, qual é o valor mínimo.

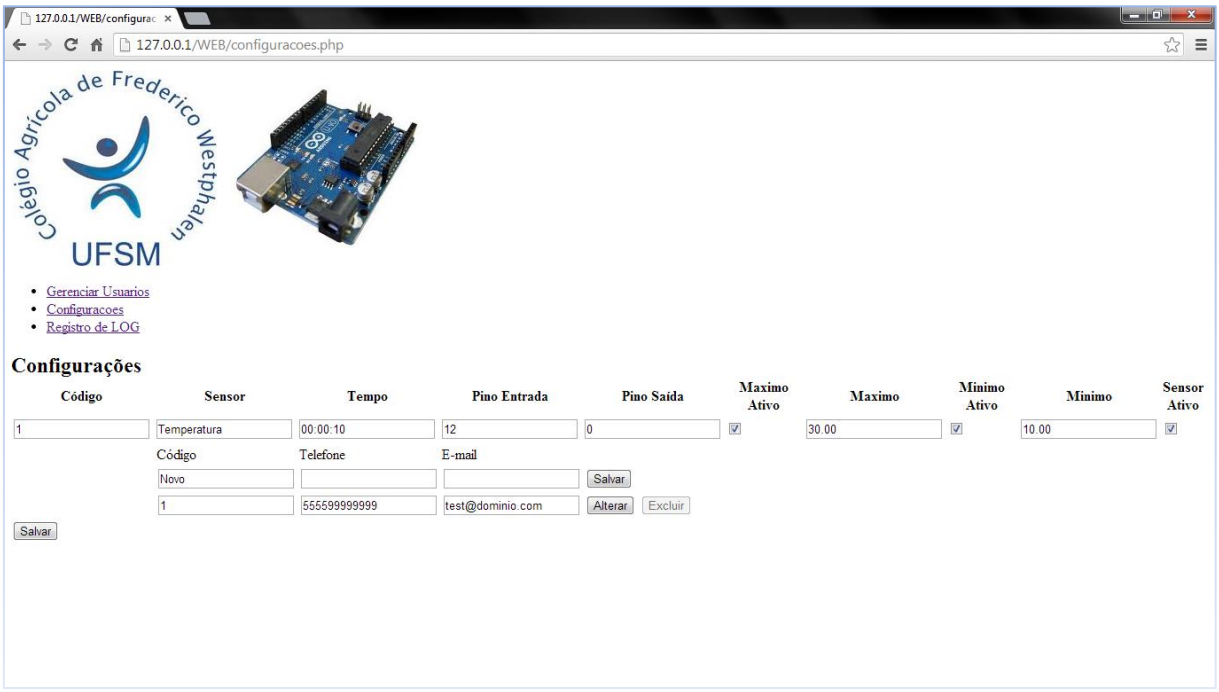

Figura 21 - Tela para configuração de sensores.

Nesta tela também serão adicionados os contatos de cada sensor, sendo que estes contatos receberão as notificações via e-mail e/ou SMS, caso o sensor ultrapassar os valores mínimos e máximos definidos.

Esta tela torna todo o sistema que está rodando no Arduino adaptativo, pois o Arduino passará a respeitar as novas configurações feitas pelo usuário administrador.

6.2.3Log de registros

Esta tela será responsável por apresentar para o usuário todos os registros de leituras do ambiente da sala de servidores. Nesta tela o usuário poderá localizar por um sensor em especifico e também por um período de data e hora.

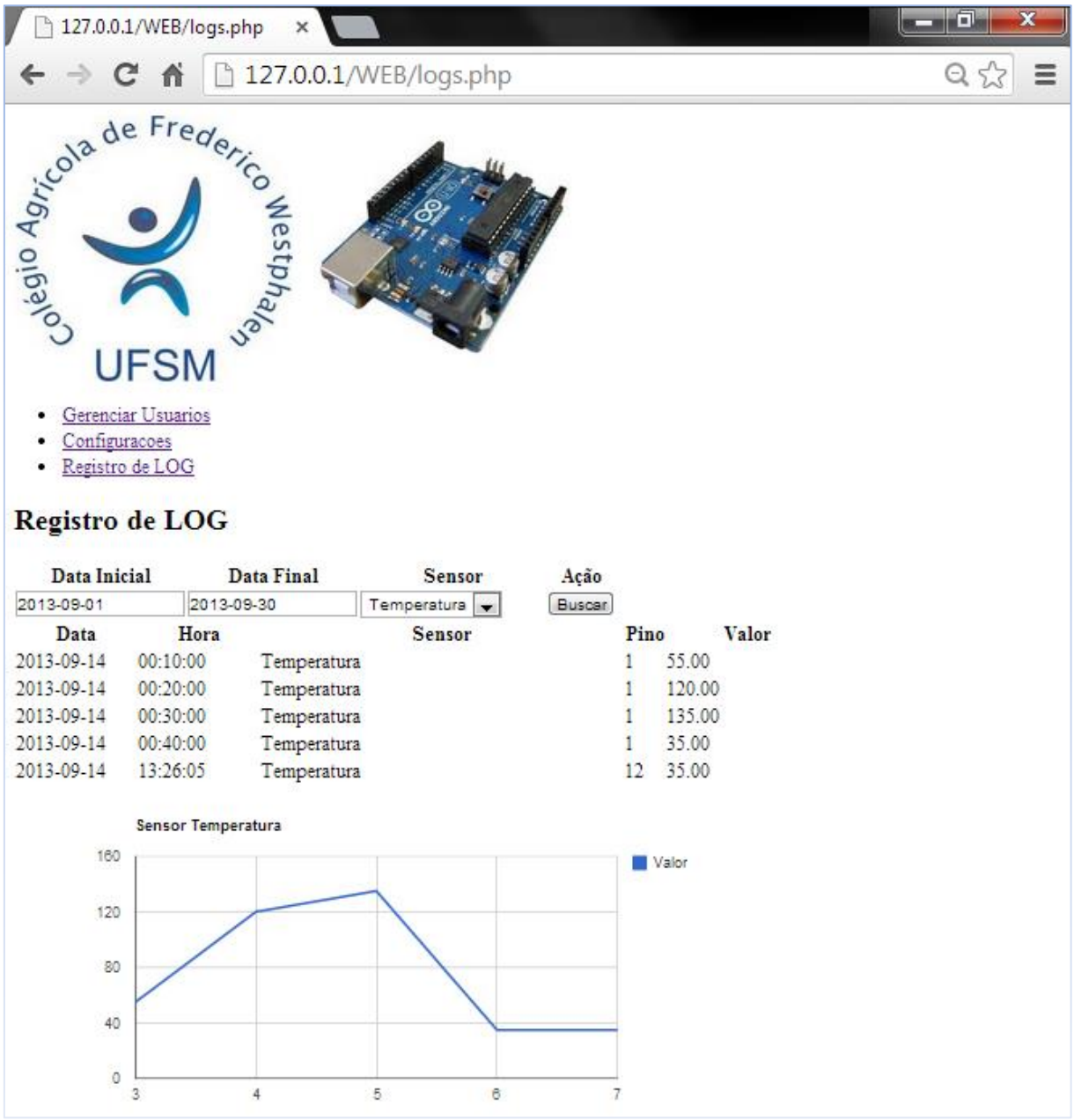

Figura 22 - Tela para consulta de log de registros.

Esta tela também irá mostrar um gráfico do sensor selecionado, demonstrando graficamente as oscilações, facilitando assim a identificação de problemas ou falhas nos equipamentos.

6.3 Segurança

Como o projeto irá controlar e gerenciar informações de grande importância para a empresa, é necessário que exista um controle de acessos dos usuários para evitar possíveis danificações, perdas de dados e uso mal intencionado da aplicação.

Para controlar os acessos de cada usuário, serão utilizados dois tipos de classificação de usuários: os convencionais e os administradores.

- **Convencionais:** Este tipo de usuário terá somente acesso à tela de log de registro, podendo somente visualizar as informações.
- **Administradores:** Além de acesso à tela de log de registro, este usuário também terá permissão de acesso à parte de gerência de usuários e configuração de sensores. Este usuário será adicionado automaticamente após ser feita a implantação do projeto e não poderá existir mais de um administrador.

Para acessar o sistema, cada usuário irá conter um login e senha, sendo que a senha será criptografada, aumentando assim o nível de segurança dos acessos.

6.4Banco de Dados

O banco de dados que utilizado é o MySQL 5.5. Serão necessárias quatro tabelas de dados, sendo elas SENSORES, SENSORESCONTATOS, LOG e USUARIOS.

6.4.1Tabela de SENSORES

Está tabela contém toda a configuração referente a cada sensor. Abaixo a definição de cada campo da tabela.

SE_CODIGO: Chave controle da tabela.

SE_DESCRICAO: Nome do sensor, exemplo “sensor de temperatura e umidade”.

SE_PINO: Pino onde o sensor esta conectado no Arduino.

SE_TEMPO: Intervalo de leitura de cada sensor.

SE_VLRMAXIMO: Valor máximo para o sensor entrar em estado de alerta.

SE_VLRMINIMO: Valor mínimo para o sensor entrar em estado de alerta.

SE_VLRMAXIMOATIVO: Valor máximo para o sensor está ativo ou não.

SE_VLRMINIMOATIVO: Valor mínimo para o sensor está ativo ou não.

SE_ATIVO: Sensor está sendo utilizado ou não.

SE_PINOSAIDA: Pino onde o LED esta conectado no Arduino.

6.4.2Tabela de SENSORESCONTATOS

Está tabela contém todos os e-mails e telefones dos responsáveis por um determinado sensor. Assim que o sensor sair as situação dos valores entre mínimo e máximo, os contatos registrados nesta tabela serão notificados. Abaixo a definição de cada campo da tabela.

SEC_CODIGO: Chave controle da tabela.

SE_CODIGO: Sensor em que o contato pertence.

SEC_EMAIL: E-mail para enviar notificações.

SEC_TELEFONE: Telefone para ser feito envio de notificações.

6.4.3 Tabela de LOG

Está tabela contem todos os registros de leitura dos sensores, sendo eles em estado de alerta ou não, para futuras analises e consultas. Abaixo a definição de cada campo da tabela.

LG_CODIGO: Chave controle da tabela.

SE_CODIGO: Sensor em que o log pertence.

LG_DATA: Data que foi feita a leitura.

LG_HORA: Hora que foi feita a leitura.

LG_PINO: Pino em que foi feita a leitura.

LG_VALOR: Valor da leitura.

6.4.4Tabela de USUARIOS

Está tabela contém todos os usuários que tem acesso a aplicação na internet, sendo que o usuário Administrador será incluído automaticamente. Abaixo a definição de cada campo da tabela.

USU_CODIGO: Chave controle da tabela.

USU_NOME: Nome do usuário.

USU_LOGIN: Login de acesso a aplicação da internet.

USU_SENHA: Senha criptografada em MD5 para acesso a aplicação da internet.

USU_TIPO: Tipo de usuário, Administrador ou Convencional.

6.4.5Modelo entidade relacionamento da base de dados

O Modelo Entidade Relacionamento, ou ER, é importante, pois com ele é possível visualizar as estruturas das tabelas, bem como as suas ligações facilitando assim a manutenção e entendimento do banco de dados.

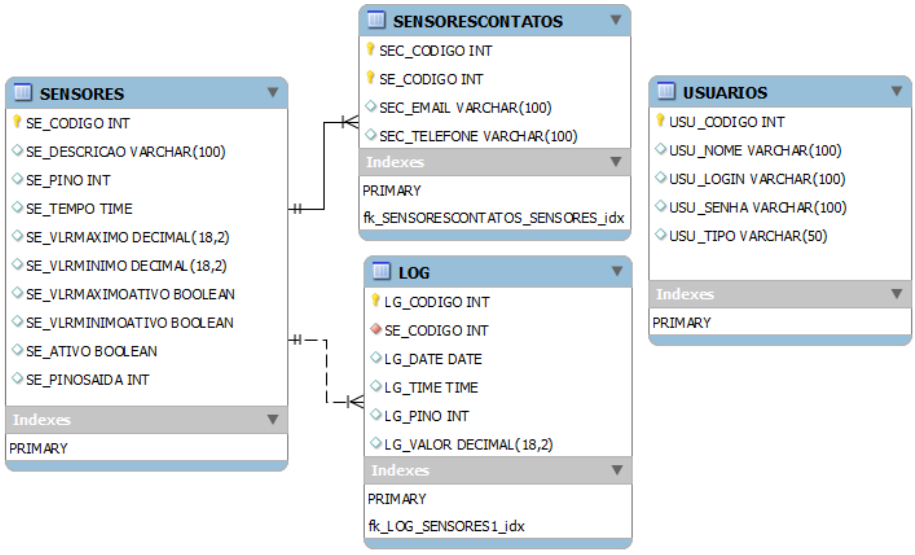

Figura 23 - Modelo entidade relacionamento (ER) do projeto.

Na figura acima é possível visualizar todas as tabelas com seus campos e seus tipos de dados, bem como o vínculo entre elas, que garantem a integridade dos dados.

7 CONCLUSÃO

Inicialmente foi realizada uma pesquisa sobre a Governança de TI, buscando esclarecer seu objetivo. Baseado nessa pesquisa ficou possível afirmar que a Governaça de TI é de responsabilidade da alta administração, e que ela deve fornecer mais controles da estrutura de TI, garantindo a sustentabilidade e a competitividade, tendo em vista que as empresas necessitam alinhar os objetivos da TI aos objetivos estratégicos da organização. Valor, risco e controle constituem a essência da governança de TI.

Trabalhou-se também um estudo sobre os modelos de referência que abordam as melhores práticas da Gestão da TI, como o CobiT e a norma ABNT NBR ISO/IEC 17799. Foram identificadas as recomendações e melhores práticas destes modelos, focadas na garantia da continuidade dos serviços e a redução da exposição do negócio aos riscos de TI.

Perante este contexto, foi desenvolvido um protótipo de hardware, composto por uma placa Arduino, com alguns *shields*, conforme pode-se verificar detalhadamente no capítulo de Desenvolvimento da Aplicação, inclusive com um fluxograma.

Além da montagem do hardware, acoplando a placa Arduino com os *Shields*, se fez necessário desenvolver outros dois softwares. Um dos softwares foi gravado no próprio Arduino, e outro, foi desenvolvido em PHP, para que os usuários pudessem interagir com o sistema. Todas as funcionalidades de software estão descritas em casos de uso, no capítulo 6.1. Também foi apresentado um diagrama entidade relacionamento do banco de dados da aplicação.

Com as etapas descritas acima alcançadas, todos os objetivos do projeto foram alcançados de forma satisfatória, tendo como produto final um protótipo ARDUINO e *Shields* acoplados e interligados a um banco de dados, com uma interface WEB intuitiva para o usuário.

Para a construção do protótipo foram encontradas diversas dificuldades sendo as principais relacionadas ao desenvolvimento da aplicação do Arduino.

A configuração da rede apresenta algumas complexidades, pois é necessário fazer a configuração de *IP*, *MAC*, *Subnet* e *Gateway*, porém após realizada a configuração não existe uma forma de verificar se a conexão foi realizada ou não, sendo necessária uma outra

máquina para enviar um pacote de dados para o Arduino para verificar se o mesmo está na rede ou não.

O envio das notificações por e-mail também foi um detalhe que tomou bastante tempo, pois existe uma precariedade de documentos que esclareçam seu funcionamento, sendo que os servidores de e-mail muitas vezes apresentam bloqueios de envio.

Outro detalhe que também foi relevante foi a busca de dados na base de dados do MySQL. Para realizar uma consulta no BD deve se ter diversos cuidados tendo em vista que a memória do Arduino é bastante limitada e também pelo o fato de ser bastante complexa a busca e conversão de dados, sendo que não existe um método que seja simples e de fácil implementação.

7.1 Trabalhos Futuros

Como sugestão e proposta de trabalhos futuros, fica aqui o desafio para criar um protótipo semelhante ao apresentando, porém, utilizando a plataforma Raspberry Pi, utilizando sua saída HDMI para mostrar dados de leituras em tempo real em um monitor, com uma interface mais atrativa.

Também fica o desafio de incluir novas funcionalidades no projeto apresentando, como por exemplo o acréscimo de controles e monitoramento do acesso de pessoas a sala dos servidores, através da utilização de *tags* RFID, e envio de SMS, com a utilização de *shield* GSM.

REFERÊNCIAS

ABNT NBR ISO/IEC 17799. (2005, 08 31). ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. *NBR ISO/IEC 17799*. Rio de Janeiro.

ARDUINO Site Oficial. (2013). *Arduino*. Retrieved 11 26, 2013, from http://www.arduino.cc/: http://www.arduino.cc/

BRAUN, D. (2010, 05 27). *HTML 5: conheça a linguagem que vai revolucionar sua navegação na web.*Retrieved 11 26, 2013, from IDGNOW: http://idgnow.uol.com.br/internet/2009/06/16/html-5-conheca-a-linguagem-que-vai-revolucionar-sua-navegacao-na-web/

BRUGNARI, A., & MAESTRELLI, L. H. (2010). Automação Residencial via WEB. *Monografia (Graduação em Engenharia da Computação) - Pontifícia Universidade Católica do Paraná (PUC, PR)*. Curitiba, PR.

CobiT 4.1. (2007). IT GOVERNANCE INSTITUTE. *CobiT 4.1 - Control Objectives for Information and related Technology*. Rolling Meadows, USA.

COBIT 4.1. (2007). IT GOVERNANCE INSTITUTE. *CobiT 4.1 - Control Objectives for Information and related Technology*. Rolling Meadows, USA.

FERNANDES, A. A. (2012). *Implantando a Governança de TI: Da estratégia à gestão dos processos e serviços 3.ed.* Rio de Janeiro: Brasport.

GASETA, E. R. (2012). *Governaça de TI* (3ª Edição ed.). Rio de Janeiro: Brasport.

ISERHARDT, M. (2010). Sistema de Automação e Controle. *Monografia (Bacharelado em Sistemas da Informação) - Universidade da Região da Campanha (URCAMP, RS)*. Bagé, RS.

MARGOLIS, M. (2011). *Arduino Cookbook.* USA: O´Reilly.

MONK, S. (2013). *Programando o Raspberry Pi* (1ª Edição ed.). São Paulo: Novatec.

NETO, R. O. (2011). Automação de iluminação residencial utilizando microcontrolador arduino e tablet ipad via wi-fi. *Monografia (Bacharelado em Engenharia da Computação) - Centro Universitário de Brasília (UniCEUB, DF)*. Brasília, DF.

O'BRIEN, J. A. (2010). *Sistemas de Informação* (3ª Edição ed.). São Paulo: Saraiva.

OLIVEIRA, W. (2012). Automação Residencial em sistema embarcado com Arduino. *Monografia (Bacharelado em Sistemas da Informação) - Faculdade Projeção*. Taguatinga, DF.

OLIVEIRO, C. A. (2001). *Faça um site - PHP4 com Bas de Dados MySQL Orientado por Projeto.* São Paulo: Érica.
PHP. (2013). *PHP*. Retrieved 08 01, 2013, from PHP: http://php.net/manual/pt_BR/intro-whatis.php

PINTO, F. D. (2010). Desenvolvimento de um Protótipo de um Sistema Domótico. *Dissertação (Mestrado em Engenharia Eletrotécnica e de Computadores) - Instituto Superior Técnico - Universidade Técnica de Lisboa*. Lisboa, Portugal.

PINTO, M. d. (2011). Aplicação de arquitetura pedagógica em curso de robótica educacional com hardware livre. *Dissertação (Pós Graduação em Informática) - Universidade Federal do Rio de Janeiro (UFRJ, RJ)*. Rio de Janeiro, RJ.
PMBOK® Guide 2004 Edition. (n.d.). PROJECT MANAGEMENT INSTITUTE. *A Guide to the Project Management Body of Knowledge*. Pennsylvania, USA.

ROBOCORE. (2013). *ROBOCORE*. Retrieved 11 26, 2013, from RoboCore Tecnologia e Empreendimentos Empresariais LTDA: www.robocore.net

SCHROEDER, R. (2012, jul/set). *CAMINHOS, Revista on-line de divulgação científica da UNIDAVI*, 25.

SILVA, J. F., & al, e. (2012, setembro 03 a 06). Construindo veículo teleoperado com Arduino para auxilio no ensino de sistemas embarcados e robótica móvel. *COBENGE XL Congresso Brasileiro de Educação em Engenharia*. Belém, PA.

VERAS, M., & TOZER, R. (2012). *Cloud Computing: nova Arquitetura de TI.* São Paulo: Brasport.

WEILL, P., & W., R. J. (2006). *Governança de TI - Tecnologia da Informação* (1ª Edição ed.). São Paulo: Books.

ANEXOS

ANEXO A – Orçamento de Aplicação Similar

ANEXO B – Código Fonte Arduino

```
#include <SPI.h>
#include <Ethernet.h>
#include <sha1.h>
#include <mysql.h>

// Dados para leituras das temperaturas, estes dados deverão ser buscados da web
int aSensorTemperatura;
float aTemperaturaMax;
float aTemperaturaMin;
String aTemperaturaEmails;
String aTemperaturaTelefones;
int aLuzTemperatura;

// Configuração da Conexão de rede
byte aMAC[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };
IPAddress aIP(10, 1, 1, 151); // ip que o arduino assumirá
byte aGATEWAY[] = { 10, 1, 1, 1 };  // ip do roteador
byte aSUBNET[] = { 255, 255, 255, 0 };

// Configuração do Banco de dados
IPAddress aBaseDadosIP(10, 1, 1, 18);
char aBaseDadosUsuario[] = "root";
char aBaseDadosSenha[] = "";
int aBaseDadosPorta = 3306;
Connector aMySQL;

// Configuracoes E-mail
EthernetClient aClient;
char aEmailServidor[] = "tcheturbo.com.br";

void setup() {
    // Sáida Serial
    Serial.begin(9600);

    // Cria/inicializa a conexão de rede
    Serial.println("Conectando na rede...");
Ethernet.begin(aMAC, aIP, aGATEWAY, aGATEWAY, aSUBNET);
delay(1000);

    // Conecta no Banco de Dados
    Serial.println("");
    Serial.println("Conectando Base de Dados...");
    Serial.print("  ");
    if      (!aMySQL.mysql_connect(aBaseDadosIP,      aBaseDadosPorta,      aBaseDadosUsuario,
aBaseDadosSenha))
        Serial.println("  Falha na conexao com o MySQL.");
```

```
    else
        Serial.println("  Conexao com MySQL realizada com sucesso.");

    // Busca as configurções de cada pino no MySQL
    fBuscaConfiguracao();

    // Defino os pinos de saída
    pinMode(aLuzTemperatura, OUTPUT);

    // Inicializo os pinos de saida
    digitalWrite(aLuzTemperatura, 0);

    // Inicio
    Serial.println("");
    Serial.println("Leituras");
}

// Função responsável por buscar a configuração de cada pino no MySQL
void fBuscaConfiguracao(){
    Serial.println("");
    Serial.println("Buscando configuracoes...");
    aMySQL.cmd_query("USE CONTROLESERVIDOR");
    aMySQL.cmd_query("SELECT     S.SE_CODIGO,     S.SE_PINO,     S.SE_TEMPO,     S.SE_VLRMAXIMO,
S.SE_VLRMINIMO,   S.SE_VLRMAXIMOATIVO,   S.SE_VLRMINIMOATIVO,   S.SE_PINOSAIDA,   SC.SEC_EMAIL,
SC.SEC_TELEFONE FROM SENSORES S LEFT JOIN SENSORESCONTATOS SC ON SC.SE_CODIGO=S.SE_CODIGO");

aMySQL.get_columns();
    row_values *vLinha = NULL;
    do {
        vLinha = aMySQL.get_next_row();
        if (vLinha != NULL) {
            aSensorTemperatura = atoi(vLinha->values[1]);
            aLuzTemperatura    = atoi(vLinha->values[7]);

            aTemperaturaMax = 999999;
            if (atoi(vLinha->values[5]) == 1)
               aTemperaturaMax = atoi(vLinha->values[3]);

aTemperaturaMin = -999999;
if (atoi(vLinha->values[6]) == 1)
               aTemperaturaMin = atoi(vLinha->values[4]);

            if (vLinha->values[8] != ""){
                if (aTemperaturaEmails != "")
                    aTemperaturaEmails = aTemperaturaEmails + ";" + vLinha->values[8];
                else
                    aTemperaturaEmails = vLinha->values[8];
            }

if (vLinha->values[9] != ""){
                if (aTemperaturaTelefones != "")
```

```
aTemperaturaTelefones = aTemperaturaTelefones + "; " + vLinha->values[9];
else
                    aTemperaturaTelefones = vLinha->values[9];
}
        }

    } while (vLinha != NULL);
    aMySQL.free_columns_buffer();
    aMySQL.free_row_buffer();

Serial.println("  Configuracao Temperatura");
Serial.print("    Pino Entrada: ");
    Serial.println(aSensorTemperatura);
    Serial.print("    Pino Saida: ");
    Serial.println(aLuzTemperatura);
    Serial.print("    Valor maxima: ");
    Serial.println(aTemperaturaMax);
    Serial.print("    Valor minimo: ");
    Serial.println(aTemperaturaMin);
Serial.print("    E-mails: ");
    Serial.println(aTemperaturaEmails);
Serial.print("    Telefones: ");
    Serial.println(aTemperaturaTelefones);
    Serial.println("Fim da busca de configuracoes.");
}

// Ativa alertas dos led/luzes
void fControlaAlertas(boolean pAlerta, int pPino){
if (pAlerta){
      digitalWrite(pPino, 255);
  }else{
digitalWrite(pPino, 0);
  }
}

void fRealizaLeitura(){
   char vMensasagem[150];

   // Busca a temperatura
   int vTemperatura = fTemperatura();
   Serial.print("  Temperatura: ");
   Serial.println(vTemperatura);

   // Grava a temperatura que foi lida
   fGravaLog(1, aSensorTemperatura, vTemperatura);

   // Verifica se a tempetura está fora dos limites configurados
   if (vTemperatura >= aTemperaturaMax ||
       vTemperatura <= aTemperaturaMin){
       fControlaAlertas(true, aLuzTemperatura);
```

```
        //Enviar a notificação ( SMS e E-mail )
        sprintf(vMensasagem, "Temperatura: %d.00", vTemperatura);
        fEnviarNotificao(vMensasagem, 1, aTemperaturaEmails, aTemperaturaTelefones);
    } else{
        fControlaAlertas(false, aLuzTemperatura);
    }
}

// Função para fazer a leitura da Temperatura
int fTemperatura(){
    return analogRead(aSensorTemperatura)/4;
}

// Grava leituras na base de dados
void fGravaLog(int pSensor, int pPino, int pValor){
    char SQLInsert[500];
    sprintf(SQLInsert, "INSERT INTO CONTROLESERVIDOR.LOG VALUES (NULL, %d, CURRENT_DATE,
CURRENT_TIME, %d, %d.00)", pSensor, pPino, pValor);
aMySQL.cmd_query(SQLInsert);
    Serial.println("    Log gravado com sucesso!");
}

void fFalhaRede(){
    byte vDados = 0;
    int vContador = 0;

aClient.println(F("QUIT"));

    while(!aClient.available()){
delay(1);
        vContador++;


        // Tenta a conexão por 5 segundos senão entede que deu Timeout
if (vContador > 5000) {
            aClient.stop();
            Serial.println("      Timeout");
        }
    }

    while(aClient.available()){
        vDados = aClient.read();
Serial.print("      ");
        Serial.println(vDados);
    }

    aClient.stop();

    Serial.println("    Desconectando do SMTP");
}
```

```
int fTimeOutConexao(){
    byte vResposta;
    byte vDados;

int vContador = 0;
    while(!aClient.available()) {
delay(1);
        vContador++;

        // Tenta a conexão por 5 segundos senão entede que deu Timeout
if (vContador > 5000) {
            aClient.stop();
            Serial.println("      Timeout");
            return 0;
        }
    }

vResposta = aClient.peek();

    Serial.print("      ");
while(aClient.available()){
vDados = aClient.read();
Serial.write(vDados);
    }

    if(vResposta >= '4'){
        fFalhaRede();
        return 0;
    }

    return 1;
}

// Função para envio de e-mail e SMS para os responsáveis
int fEnviarNotificao(char vMensagem[150], int pSensor, String pEmails, String pTelefones){
    if (pEmails != ""){
        Serial.print("    Enviando notificacao para ");
Serial.println(pEmails);

        boolean vPassa = true;

Serial.println("      Incializando as configuracoes de STMP...");
vPassa = aClient.connect(aEmailServidor, 587);
        if (vPassa)
Serial.println("      Conexao com STMP realizada com sucesso.");
        else
            Serial.println("      Falha na conexao com o STMP.");

        if (vPassa)
            vPassa = fTimeOutConexao() == 1;
        if (vPassa)
```

```
            aClient.println(F("helo 187.108.17.233"));

if (vPassa)
            vPassa = fTimeOutConexao() == 1;
        if (vPassa)
            aClient.println(F("MAIL From: <nfe@impactosistemas.com.br>"));

        if (vPassa)
            vPassa = fTimeOutConexao() == 1;
        if (vPassa){
            aClient.print(F("RCPT To: <"));
            aClient.print(pEmails);
            aClient.println(F(">"));
        }

        if (vPassa)
            vPassa = fTimeOutConexao() == 1;
        if (vPassa)
            aClient.println(F("DATA"));

        if (vPassa)
            vPassa = fTimeOutConexao() == 1;
        if (vPassa){
            aClient.print(F("To: You <"));
            aClient.print(pEmails);
            aClient.println(F(">"));
            aClient.println(F("From: Arduino <nfe@impactosistemas.com.br>"));
aClient.println(F("Subject: Servidor - Notificação\t\r\n"));
            aClient.println(vMensagem);
            aClient.println(F(""));
            aClient.println(F("Mensagem gerada automaticamente pelo Arquino que controla o
servidor"));
aClient.println(F("."));
        }

        if (vPassa)
            vPassa = fTimeOutConexao() == 1;
        if (vPassa)
            aClient.println(F("QUIT"));

aClient.stop();

if (vPassa)
            Serial.println("    Envio de notificacao concluido com sucesso.");
        else
            Serial.println("    Falha no envio de notificacao.");
    }

    Serial.println("");
}
void loop(){
```

```
    fRealizaLeitura();
    delay(1000);
}
```

ANEXO C – Código da Aplicação

configuracao.php

```
<?php
	define("SERVIDOR", "127.0.0.1");
	define("USUARIO", "root");
	define("SENHA", "");
	define("BANCO", "CONTROLESERVIDOR");
?>
```

verificalogin.php

```
<?php
	ob_start();
	session_start();
	if($_SESSION['Logado'] != 'S') {
		header('location:login.php');
	}
?>
```

login.php

```
<?php
	include("menu.php");
	include("funcoes.php");
	ob_start();
	session_start();
	$_SESSION['Logado'] = 'N';
	$_SESSION['Tipo']   = '';
	if($_SERVER["REQUEST_METHOD"] == "POST") {
		$vLogin = $_POST["edlogin"];
		$vSenha = $_POST["edsenha"];
		AbreConexao();
		$vConsulta = mysql_query("SELECT USU_TIPO FROM USUARIOS U WHERE USU_LOGIN =
'$vLogin' AND USU_SENHA = '$vSenha'");
		while($vLinha = mysql_fetch_array($vConsulta)){
			$_SESSION['Logado'] = 'S';
			$_SESSION['Tipo']   = $vLinha["USU_TIPO"];

			header('location:index.php');
		}
	}
?>
<form action="login.php" method="post">
	<label>Login</label>
	<input autofocus="true" name="edlogin">
	<label>Senha</label>
	<input type="password" name="edsenha">
	<input id="submit" type="submit" value="Logar" name="submit">
```

```
</form>
```

funcoes.php

```
<?php
	include("configuracao.php");
	function AbreConexao(){
		@mysql_connect(SERVIDOR, USUARIO, SENHA) or die(mysql_error());
		@mysql_select_db(BANCO) or die(mysql_error());
	}
	function FechaConexao(){
		@mysql_close;
	}
?>
```

index.php

```
<?php
	require("verificalogin.php");
	include("menu.php");
?>
<!DOCTYPE HTML>
<head>
</head>
<body>
</body>
```

menu.php

```
<table>
	<tr>
		<td><img id="logo" src="cafw.jpg" alt="CAFW"></td>
		<td align="right"><img id="logo" src="arduino.jpg" alt="ARDUNO"></td>
	</tr>
</table>
<?php
	if($_SESSION['Tipo'] == 'Administrador') {
		echo'
		<nav>
			<ul>
				<li><a href="usuarios.php">Gerenciar Usuarios</a></li>
				<li><a href="configuracoes.php">Configuracoes</a></li>
				<li><a href="logs.php">Registro de LOG</a></li>
			</ul>
		</nav>';
	}
	else
	if($_SESSION['Tipo'] == 'Convencional') {
		echo'
		<nav>
			<ul>
				<li><a href="logs.php">Registro de LOG</a></li>
			</ul>
```

```
		</nav>';
	}else{
		echo'
		<nav>
			<ul>
				<li><a href="login.php">Login</a></li>
			</ul>
		</nav>';
	}
?>
```

usuarios.php

```
<?php
	require("verificalogin.php");
	include("menu.php");
	include("funcoes.php");
	AbreConexao();
	$vMensagem  = "";
	$vSaida     = "";
	if ($_SERVER['REQUEST_METHOD'] == "POST"){
		$vNome  = $_POST['edNovoNome'];
		$vLogin = $_POST['edNovoLogin'];
		$vSenha = $_POST['edNovoSenha'];
		if ($vNome != ""){// Cadastrando novo
			@mysql_query("INSERT INTO USUARIOS VALUES (  NULL , '$vNome', '$vLogin',
'$vSenha', 'Convencional'); ");
		}
		for ($i=1; $i <= $_POST['edQuantidadeLinhas']; $i++) {
			$vCodigo = $_POST['edCodigo' . $i];
			$vNome   = $_POST['edNome' . $i];
			$vSenha  = $_POST['edSenha' . $i];
			@mysql_query("UPDATE  USUARIOS  SET  USU_NOME  =  '$vNome',  USU_SENHA  =
'$vSenha' WHERE USU_CODIGO = '$vCodigo'; ");
		}
	}
	// Carrega dados de consulta
	$vConsulta = mysql_query("SELECT * FROM USUARIOS ORDER BY 2");
	$i         = 0;
	while($vLinha = mysql_fetch_array($vConsulta)){
		$i          = $i + 1;
		$vCodigo    = $vLinha["USU_CODIGO"];
		$vNome      = $vLinha["USU_NOME"];
		$vLogin     = $vLinha["USU_LOGIN"];
		$vSenha     = $vLinha["USU_SENHA"];
		$vSaida = $vSaida .
			"<tr>
				<td>
					<input name='edCodigo" . $i . "' id='edCodigo" . $i . "'
type='text' value='" . $vCodigo . "' readonly/>
				</td>
				<td>
```

```
                                <input name='edNome" . $i . "' id='edNome" . $i . "'
type='text' value='" . $vNome . "' readonly/>
                        </td>
                        <td>
                                <input name='edLogin" . $i . "' id='edLogin" . $i . "'
type='text' value='" . $vLogin . "' readonly/>
                        </td>
                        <td>
                                <input name='edSenha" . $i . "' id='edSenha" . $i . "'
type='password' value='" . $vSenha . "' readonly/>
                        </td><td>";
        if ($vCodigo != 1){
                $vSaida = $vSaida . "
                        <input type='button' value='Alterar' onClick='fEditar(" .
$i . ")'/>
                        <input disabled type='button' value='Excluir'/>";
        }
        $vSaida = $vSaida . "</td></tr>";
    }
?>
<!DOCTYPE HTML>
<head>
    <script type="text/javascript">
        function fEditar(i){
            document.getElementById('edNome' + i).readOnly  = false;
            document.getElementById('edSenha' + i).readOnly = false;
        }
    </script>
</head>
<body>
    <section>
        <h1>Usuários</h1>
            <form action="usuarios.php" method="post">
                <label><?php echo $vMensagem; ?></label>
                <input  hidden  name="edQuantidadeLinhas"  id="edQuantidadeLinhas"
type="text" value="<?php echo $i; ?>" />
                <table>
                    <th>
                        <label>Código</label>

                    </th>
                    <th>
                        <label>Nome</label>
                    </th>
                    <th>
                        <label>Login</label>
                    </th>
                    <th>
                        <label>Senha</label>
                    </th>
                    <th>
```

```
                                        <label>Ação</label>
                                    </th>
                                    <tr>
                                        <td>
                                            <input name="edNovoCodigo" value="Novo">
                                        </td>
                                        <td>
                                            <input  name='edNovoNome'  id='edNovoNome'
type='text' value=''/>
                                        </td>
                                        <td>
                                            <input  name='edNovoLogin'  id='edNovoLogin'
type='text' value=''/>
                                        </td>
                                        <td>
                                            <input  name='edNovoSenha'  id='edNovoSenha'
type='password' value=''/>
                                        </td>
                                        <td>
                                            <input  type="submit"  class='BtAdicionar'
value="Salvar"/>
                                        </td>
                                    </tr>
                                    <?php echo $vSaida; ?>

                                </table>
                                <input type="submit" class='BtSalvar' value="Salvar"/>
                            </form>
        </section>
</body>
```

configuracoes.php

```
<?php
        require("verificalogin.php");
        include("menu.php");
        include("funcoes.php");
        AbreConexao();
        if ($_SERVER['REQUEST_METHOD'] == "POST"){
                $vCodigo      = $_POST['edCodigo1'];
                $vPino        = $_POST['edPino1'];
                $vPinoSaida   = $_POST['edPinoSaida1'];
                $vTempo       = $_POST['edTempo1'];
                $vMaximo      = $_POST['edMaximo1'];
                $vMaximoAtivo = $_POST['cbMaximoAtivo1'];
                $vMinimo      = $_POST['edMinimo1'];
                $vMinimoAtivo = $_POST['cbMinimoAtivo1'];
                $vAtivo       = $_POST['cbAtivo1'];
                @mysql_query("UPDATE SENSORES SET SE_PINO = '$vPino',
                                                  SE_PINOSAIDA = '$vPinoSaida',
                                                  SE_TEMPO = '$vTempo',
                                                  SE_VLRMAXIMO = '$vMaximo',
```

```
                                                SE_VLRMINIMO = '$vMinimo',
                                                SE_VLRMAXIMOATIVO = '$vMaximoAtivo',
                                                SE_VLRMINIMOATIVO = '$vMinimoAtivo',
                                                SE_ATIVO = '$vAtivo'
                                                WHERE SE_CODIGO = '$vCodigo'; ");
                if (($vTelefone . $vEmail) != ""){// Cadastrando novo
                        @mysql_query("INSERT    INTO    SENSORESCONTATOS    VALUES    ((SELECT
COALESCE(MAX(S.SEC_CODIGO), 0) + 1 FROM SENSORESCONTATOS S WHERE S.SE_CODIGO = $vCodigo),
'$vCodigo', '$vEmail  ', '$vTelefone'); ");
                }
                for ($i=1; $i <= $_POST['edContatos']; $i++) {
                        $vItem  = $_POST['edContCod' . $i];
                        $vFone  = $_POST['edContFone' . $i];
                        $vEmail = $_POST['edContEmail' . $i];
                        @mysql_query("UPDATE  SENSORESCONTATOS  SET  SEC_TELEFONE  =  '$vFone',
SEC_EMAIL = '$vEmail' WHERE SEC_CODIGO = '$vItem' AND SE_CODIGO = '$vCodigo'; ");
                }
        }
        $vMensagem  = "";
        $vSaida     = "";
        // Carrega dados de consulta
        $i         = 0;
        $vConsulta = mysql_query("SELECT S.* FROM SENSORES S ORDER BY 2");
        while($vLinha = mysql_fetch_array($vConsulta)){
                $i            = $i + 1;
                $vCodigo      = $vLinha["SE_CODIGO"];
                $vSensor      = $vLinha["SE_DESCRICAO"];
                $vTempo       = $vLinha["SE_TEMPO"];
                $vPino        = $vLinha["SE_PINO"];
                $vPinoSaida   = $vLinha["SE_PINOSAIDA"];
                $vMaximo      = $vLinha["SE_VLRMAXIMO"];
                $vMinimo      = $vLinha["SE_VLRMINIMO"];
                $vMaximoAtivo = $vLinha["SE_VLRMAXIMOATIVO"];
                $vMinimoAtivo = $vLinha["SE_VLRMINIMOATIVO"];
                $vAtivo       = $vLinha["SE_ATIVO"];
                $vSaida = $vSaida .
                        "<tr>
                                <td>
                                        <input name='edCodigo" . $i . "' id='edCodigo" . $i . "'
type='text' value='" . $vCodigo . "'/>
                                </td>
                                <td>
                                        <input name='edSensor" . $i . "' id='edSensor" . $i . "'
type='text' value='" . $vSensor . "'/>
                                </td>
                                <td>
                                        <input name='edTempo" . $i . "' id='edTempo" . $i . "'
type='text' value='" . $vTempo . "'/>
                                </td>
                                <td>
```

```
                                    <input name='edPino" . $i . "' id='edPino" . $i . "'
type='text' value='" . $vPino . "'/>
                                </td>
                                <td>
                                        <input name='edPinoSaida" . $i . "' id='edPinoSaida" . $i
. "' type='text' value='" . $vPinoSaida . "'/>
                                </td>
                                <td>
                                        <input name='cbMaximoAtivo" . $i . "' id='cbMaximoAtivo"
. $i . "' type='checkbox' " . ($vMaximoAtivo==1 ? 'checked' : '') . " value='1' />

                                </td>
                                <td>
                                        <input name='edMaximo" . $i . "' id='edMaximo" . $i . "'
type='text' value='" . $vMaximo . "'/>
                                </td>
                                <td>
                                        <input name='cbMinimoAtivo" . $i . "' id='cbMinimoAtivo"
. $i . "' type='checkbox' " . ($vMinimoAtivo==1 ? 'checked' : '') . " value='1' />

                                </td>
                                <td>
                                        <input name='edMinimo" . $i . "' id='edMinimo" . $i . "'
type='text' value='" . $vMinimo . "'/>
                                </td>
                                <td>
                                        <input name='cbAtivo" . $i . "' id='cbAtivo" . $i . "'
type='checkbox' " . ($vAtivo==1 ? 'checked' : '') . " value='1' />

                                </td>
                        </tr>";
                        $vSaida = $vSaida . '<tr><td></td><td>Código</td><td>Telefone</td><td>E-
mail</td>';
                        $vSaida = $vSaida . "
                                        <tr>
                                                <td>
                                                        <input  hidden  name='edNovo'  id='edNovo'
type='text' value='" . $vCodigo . "' readonly/>
                                                </td>
                                                <td>
                                                        <input name='edNovoText' value='Novo'>
                                                </td>
                                                <td>
                                                        <input                name='edNovoTelefone'
id='edNovoNome' type='text' value=''/>
                                                </td>
                                                <td>
                                                        <input name='edNovoEmail' id='edNovoLogin'
type='text' value=''/>
                                                </td>
                                                <td>
```

```
                                                    <input  type='submit'  class='BtAdicionar'
value='Salvar'/>
                                                </td>
                                        </tr>";
                // Carrega dados de consulta
                $vContatos  =  mysql_query("SELECT  SC.*  FROM  SENSORESCONTATOS  SC  WHERE
SC.SE_CODIGO = " . $vCodigo . " ORDER BY 2, SC.SEC_CODIGO");
                $j = 0;
                while($vDados = mysql_fetch_array($vContatos)){
                        $j = $j + 1;
                        $vFone   = $vDados["SEC_TELEFONE"];
                        $vEmail  = $vDados["SEC_EMAIL"];
                        $vSaida = $vSaida . '</tr>';
                        $vSaida = $vSaida . "<tr>
                                <td>
                                        <input hidden name='edOrigem" . $j . "' id='edOrigem" .
$j . "' type='text' value='" . $vCodigo . "' readonly/>
                                </td>
                                <td>
                                        <input name='edContCod" . $j . "' id='edContCod" . $j .
"' type='text' value='" . $vDados["SEC_CODIGO"] . "' readonly/>
                                </td>
                                <td>
                                        <input name='edContFone" . $j . "' id='edContFone" . $j .
"' type='text' value='" . $vFone . "' readonly/>
                                </td>
                                <td>
                                        <input name='edContEmail" . $j . "' id='edContEmail" . $j
. "' type='text' value='" . $vEmail . "' readonly/>
                                </td>
                                <td>
                                        <input type='button' value='Alterar' onClick='fEditar(" .
$j . ")'/>
                                        <input disabled type='button' value='Excluir'/>
                                </td>";
                }
                $vSaida  =  $vSaida  .  "</tr><input  hidden  name='edContatos'  id='edContatos'
type='text' value='" . $j . "' readonly/>";
        }
?>
<!DOCTYPE HTML>
<head>
        <script type="text/javascript">
                function fEditar(i){
                        document.getElementById('edContFone' + i).readOnly  = false;
                        document.getElementById('edContEmail' + i).readOnly = false;
                }
        </script>
</head>
<body>
        <section>
```

```
            <h1>Configurações</h1>
                <form action="configuracoes.php" method="post">
                    <table>
                        <th>
                            <label>Código</label>

                        </th>
                        <th>
                            <label>Sensor</label>

                        </th>
                        <th>
                            <label>Tempo</label>
                        </th>
                        <th>
                            <label>Pino Entrada</label>

                        </th>
                        <th>
                            <label>Pino Saída</label>

                        </th>
                        <th>
                            <label>Maximo Ativo</label>

                        </th>
                        <th>
                            <label>Maximo</label>
                        </th>
                        <th>
                            <label>Minimo Ativo</label>

                        </th>
                        <th>
                            <label>Minimo</label>
                        </th>
                        <th>
                            <label>Sensor Ativo</label>
                        </th>
                        <?php echo $vSaida; ?>

                    </table>
                    <input type="submit" class='BtSalvar' value="Salvar"/>
                </form>
        </section>
</body>
```

logs.php

```
<?php
        require("verificalogin.php");
        include("menu.php");
```

```
include("funcoes.php");
AbreConexao();
$vFiltros    = "";
$vMensagem   = "";
$vSaida      = "";
$vCombobox   = "<select name='cbSensor' id='cbSensor'>";
$vGrafico    = "";
$vDataInicio   = "2013-11-01";
$vDataFim        = "2013-11-30";
$vSensorFiltro = 1;
if ($_SERVER['REQUEST_METHOD'] == "POST"){
        $vDataInicio   = $_POST['edDtInicio'];
        $vDataFim      = $_POST['edDtFim'];
        $vSensorFiltro = $_POST['cbSensor'];
}
if ($vDataInicio != "")
        $vFiltros = $vFiltros . " AND L.LG_DATE >= '" . $vDataInicio . "' ";
if ($vDataFim != "")
        $vFiltros = $vFiltros . " AND L.LG_DATE <= '" . $vDataFim . "' ";
if ($vSensorFiltro != "")
        $vFiltros = $vFiltros . " AND S.SE_CODIGO = '" . $vSensorFiltro . "' ";
// Carrega dados de consulta
$vConsulta = mysql_query("SELECT L.*, S.SE_DESCRICAO FROM LOG L INNER JOIN SENSORES S
ON S.SE_CODIGO = L.SE_CODIGO WHERE L.LG_CODIGO > 0 " . $vFiltros . " ORDER BY 3, 4, 2");

$i         = 0;
while($vLinha = mysql_fetch_array($vConsulta)){
        $i       = $i + 1;
        $vLog    = $vLinha["LG_CODIGO"];
        $vData   = $vLinha["LG_DATE"];
        $vHora   = $vLinha["LG_TIME"];
        $vSensor = $vLinha["SE_DESCRICAO"];
        $vPino   = $vLinha["LG_PINO"];
        $vValor  = $vLinha["LG_VALOR"];
        $vSaida = $vSaida .
                "<tr>
                        <td width='15%'>$vData</td>
                        <td width='15%'>$vHora</td>
                        <td width='45%'>$vSensor</td>
                        <td width='5%'>$vPino</td>
                        <td width='20%'>$vValor</td>
                </tr>";
        $vGrafico = $vGrafico . ",
          [$vLog, $vValor]";
}
if ($vSaida == ""){
        $vSaida = $vSaida .
                "<tr>
                        <td width='15%'></td>
                        <td width='15%'></td>
                        <td width='45%'>Nenhuma leitura localizada.</td>
```

```
                        <td width='5%'></td>
                        <td width='20%'></td>
                    </tr>";
        }
        if ($vGrafico == ""){
                $vGrafico = $vGrafico . ', [0, 0], [0, 0]';
        }
        // Monta a combobox dos pinos que podem ser escolhidos
        $vConsulta = mysql_query("SELECT SE_CODIGO, SE_DESCRICAO FROM SENSORES L WHERE SE_ATIVO
= 1 ORDER BY 2");
        while($vLinha = mysql_fetch_array($vConsulta)){
                if ($vLinha["SE_CODIGO"] == $vSensorFiltro)
                        $vCombobox = $vCombobox . "<option value=" . $vLinha["SE_CODIGO"] . "
selected>" . $vLinha["SE_DESCRICAO"] . "</option>";
                else
                        $vCombobox = $vCombobox . "<option value=" . $vLinha["SE_CODIGO"] . ">"
. $vLinha["SE_DESCRICAO"] . "</option>";
        }
        $vCombobox = $vCombobox . "</select>";
?>
<!DOCTYPE HTML>
<head>
<script type="text/javascript" src="https://www.google.com/jsapi"></script>
<script type="text/javascript">
      google.load("visualization", "1", {packages:["corechart"]});
      google.setOnLoadCallback(drawChart);
      function drawChart() {
var data = google.visualization.arrayToDataTable([
          ['Leitura', 'Valor']
<?php echo $vGrafico; ?>
        ]);
        var options = {
title: 'Sensor Temperatura'
        };
        var chart = new google.visualization.LineChart(document.getElementById('chart_div'));
        chart.draw(data, options);
      }
</script>
</head>
<body>
        <section>
                <h1>Registro de LOG</h1>
                        <form action="logs.php" method="post">
                                <table width="500px">
                                        <th width='20%'>
                                                <label>Data Inicial</label>

                                        </th>
                                        <th width='20%'>
                                                <label>Data Final</label>
```

```
                                        </th>
                                        <th width='50%'>
                                                <label>Sensor</label>
                                        </th>
                                        <th width='10%'>
                                                <label>Ação</label>
                                        </th>
                                        <tr>
                                                <td>
                                                        <input   name='edDtInicio'   id='edDtInicio'
value='<?php echo $vDataInicio; ?>'>
                                                </td>
                                                <td>
                                                        <input      name='edDtFim'      id='edDtFim'
type='text' value='<?php echo $vDataFim; ?>'/>
                                                </td>
                                                <td>
                                                        <?php echo $vCombobox; ?>
                                                </td>
                                                <td>
                                                        <input    type="submit"    class='btBuscar'
value="Buscar"/>
                                                </td>
                                        </tr>
                                </table>
                        <div id="chart_div" style="width: 600px; height: 300px;"></div>
                                <table>
                                        <th>
                                                <label>Data</label>

                                        </th>
                                        <th>
                                                <label>Hora</label>
                                        </th>
                                        <th>
                                                <label>Sensor</label>
                                        </th>
                                        <th>
                                                <label>Pino</label>
                                        </th>
                                        <th>
                                                <label>Valor</label>
                                        </th>

                                        <?php echo $vSaida; ?>

                                </table>
                        <html>
                </form>
        </section>
</body>
```