

ESTUDO DE CASO: UTILIZAÇÃO DO ARDUINO PARA UM SISTEMA DE CONTROLE REMOTO DE DISPOSITIVOS VIA INTERNET

Guilherme C. A. Tolentino, Douglas B. Tsukamoto, Shigueo Nomura
Universidade Federal de Uberlândia, Faculdade de Engenharia Elétrica, Uberlândia – MG,
E-mail: guilhermecostagca@gmail.com, douglasbtsukamoto@gmail.com, shigueonomura@feelt.ufu.br

Resumo – Este trabalho propõe o desenvolvimento de um sistema de monitoramento da temperatura de um ambiente capaz de alertar o usuário via Internet quando esta temperatura exceder o limite estabelecido. E ainda, o sistema deverá ser capaz de enviar comandos do usuário via Internet ao equipamento de controle da temperatura desse ambiente. O sistema consiste de uma integração do *hardware* representado por uma plataforma de prototipagem eletrônica (constituída pelo dispositivo Arduino Uno e pelo módulo Ethernet Shield) com o *software* responsável pela comunicação de dados via Internet. O funcionamento do produto final foi devidamente comprovado através de testes mostrando que o sistema é capaz de atingir os objetivos propostos. Além disso, o trabalho permitiu verificar que a ferramenta adotada (Arduino) é versátil e de fácil utilização para se desenvolver sistemas de baixo custo e úteis como o projeto proposto.

Palavras-Chave – Arduino, Controle Remoto, Dispositivo, Ethernet Shield, Internet.

CASE STUDY: ARDUINO USE FOR A REMOTE CONTROL SYSTEM OF DEVICES VIA INTERNET

Abstract – This work proposes the development of a monitoring system for environmental temperature that is able to alert the user via the Internet in case of exceeded temperature. Also, the system should be able to send user commands via the Internet to the equipment for temperature control of this environment. The system consists of a integration between the hardware represented by a prototyping platform (constituted by Arduino Uno electronic device and Ethernet Shield module) and the software that is responsible for data communication via the Internet. The operation of the final product was duly proven through tests showing that the system is able to achieve the goal of this work. Furthermore, the work showed that the adopted tool (Arduino) is versatile and easy for use in developing low-cost and useful systems such as the proposed project.

Keywords – Arduino, Device, Ethernet Shield, Internet, Remote Control.

I. INTRODUÇÃO

Na medida em que o uso de sistemas computacionais prolifera na sociedade torna-se comum o uso de aplicações em tempo real [1]. Busca-se cada vez mais fazer com que a troca e o processamento das informações de um sistema sejam eficientes e rápidos [1]. Muitas vezes, é necessário que um sistema seja controlado à distância tanto para que custos com manutenção sejam diminuídos quanto para se obterem respostas instantâneas a problemas ou a qualquer evento ocorrido [1]. Assim como acontece na área de automação e controle, em sites que enviam respostas automáticas aos usuários, há necessidade de monitoramento e envio rápido de respostas. É necessário buscar mais velocidade nos diversos processos, procurando minimizar os custos provenientes da logística para manutenção do sistema e os investimentos aplicados para um benefício maior.

Ao observar a evolução humana, nota-se que o conhecimento adquirido e as tecnologias desenvolvidas têm sido aplicados na adaptação do homem ao ambiente em que está inserido. Por exemplo, num ambiente que envolve a transposição de grandes distâncias para a troca de informações, tem-se tentado o uso do controle remoto. Pode-se dizer que a ciência se desenvolve a fim de melhorar em algum aspecto a vida humana, seja em termos de conforto como também em termos de minimização de custos. A engenharia também se ocupa de desenvolver equipamentos, dispositivos e mecanismos que facilitam a vida das pessoas.

Ainda com relação ao ambiente em que vive, tendo em vista que o clima é um fator decisivo na maneira de viver em qualquer região, já que irá ditar desde o modo que as pessoas se vestem até a forma de se trabalhar, o homem precisou se adequar ao clima e muitas vezes “controlá-lo”. Sendo assim, foram desenvolvidos aparelhos capazes de controlar a temperatura em um ambiente. Um exemplo desse tipo de aparelho é o ar condicionado que está sendo largamente utilizado para o condicionamento do ambiente. O funcionamento do aparelho precisa ser controlado pela programação da temperatura por um controle remoto ou diretamente no próprio aparelho.

A necessidade de um ambiente climatizado para o trabalho não se restringe aos seres humanos. Os equipamentos eletrônicos tais como computadores (servidores) precisam de um ambiente com temperatura controlada. Assim, numa sala de servidores em que os computadores precisam ficar ligados ininterruptamente e não há presença humana, torna-se imprescindível que haja um monitoramento à distância da temperatura desse ambiente.



XI CEEL – ISSN 2178-8308
25 a 29 de novembro de 2013
Universidade Federal de Uberlândia – UFU
Uberlândia – Minas Gerais – Brasil

dos conversores Analógico-Digital (A/D) do Arduino para se fazer o acoplamento.

Em termos de circuito, esse conversor tem entrada máxima de 5 V. Como ele é de 10 bits, consegue diferenciar 1024 (de 0 a 1023) valores entre 0 V e 5 V. Ou seja, quando a leitura for de 5 V na entrada, o valor será convertido para 1023. Assim, como no exemplo abaixo, é possível de se obter qualquer valor através de uma interpolação simples:

$$\begin{aligned} 1023 & \text{---} 5\text{V} \\ 512 & \text{---} x \\ x & = (5 * 512) / 1023 = 2.5 \text{ V} \end{aligned}$$

Ou seja, se o conversor A/D mostrar o valor digital 512 significa que está entrando uma tensão de 2.5 V.

Sabe-se que o LM35 é um sensor de temperatura linear que fornece 10 mV para cada grau Celsius que for medido.

O Arduino Uno se comunica com o computador tendo o Ethernet Shield como intermediário. A comunicação entre Arduino e Shield é do tipo SPI (*Serial Peripheral Interface*), na qual o primeiro se comporta como mestre e o segundo como escravo.

O pino 13 do Arduino é o responsável pelo envio de um sinal de pulso cronometrado chamado SCK (*Clock Signal*) para manter a sincronização entre as duas placas para envio e recebimento de dados. O pino 10 do Arduino é o CS (*Chip Select*), que controla quando as duas placas irão se comunicar da seguinte forma: quando o pino CS está em nível lógico baixo (0V) ocorre comunicação entre os dois e quando em nível lógico alto (5V) não há comunicação entre os dois. O pino 11 do Arduino é o SI (*Slave Input*), responsável pelo envio de dados do Arduino para o Shield. O pino 12 é o SO (*Slave Output*), responsável por enviar a resposta do Shield para o Arduino. O Shield Ethernet é alimentado pelo Arduino com uma tensão Vcc de 3.3 V. A Figura 4 apresenta uma ilustração da comunicação entre o Arduino e o Ethernet Shield.

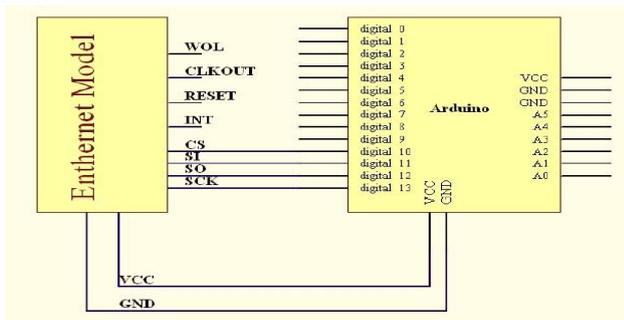


Fig. 4. Acoplamento Ethernet – Arduino

A Figura 5 apresenta uma foto do *hardware* montado para o projeto em questão conectado a um computador via cabo de Ethernet.

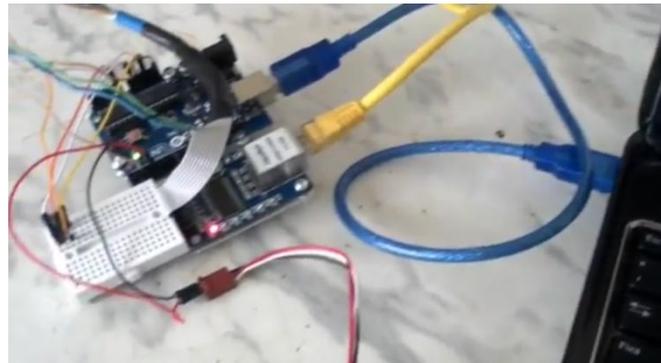


Fig. 5. Montagem do *hardware* do sistema proposto

B. Software

O Arduino foi configurado de forma que a página em HTML possa ser criada e nela sejam mostrados os dados da temperatura atual e um botão para ligar ou desligar o *cooler*. Primeiramente, configuramos o IP e MAC com um endereço que não esteja sendo utilizado por outros equipamentos.

Utilizando-se o *buffer* para análise de entrada e saída de dados dos dispositivos, criou-se a página em HTML com a função “print_webpage” que mostra um cabeçalho padrão e a página contendo a temperatura e o botão do *cooler*. Há também uma função “getCurrentTemp” que transforma o valor lido no sensor de temperatura em um número e em seguida converte o valor numérico para *string* para que seja mostrado na tela.

Uma função criada no “GoBetWino” acessa no navegador padrão a página “localhost/sendmail.php”. O *localhost* é criado a partir do programa Wamp, que instala o Apache, PHP e MySQL [7]. O arquivo “sendmail.exe” é o que envia o e-mail. A sua configuração é feita em “sendmail.ini” [8] com o servidor SMTP, o e-mail e senha de origem, o e-mail de destino, a porta do envio do e-mail e a criptografia utilizada. O envio do e-mail acontece quando se acessa em seu navegador padrão: “localhost/sendmail.php” [9]. No “sendmail.php” há o e-mail do destino, uma função mail que envia o e-mail, o texto do e-mail e uma função booleana que retorna se o e-mail foi enviado ou não.

A Figura 6 apresenta um diagrama simplificado das classes envolvidas para o desenvolvimento do *software*.

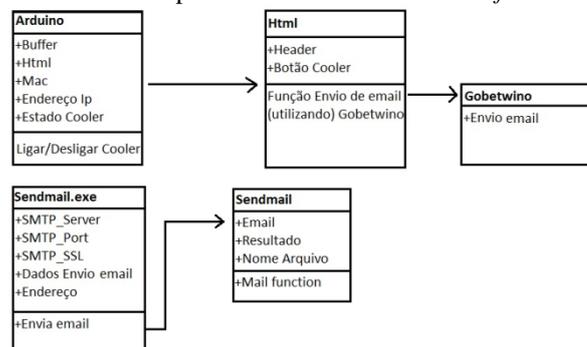


Fig. 6. Representação gráfica simples do diagrama de classes

O sensor é conectado no Arduino com 5 V, GND, entrada analógica 0. Como foi descrito anteriormente, o sensor fornece ao Arduino uma tensão diretamente proporcional à temperatura. O *cooler* é ligado no GND e em uma saída digital, a de número 4. A saída fornece sempre nível alto ou

baixo e muda sempre que se aperta o botão na página em HTML. A função desse botão muda de acordo com o estado (ligado ou desligado) do *cooler*. Vale destacar que o Shield é o intermediário entre a comunicação do Arduino com o computador. Quando o computador requisitar o endereço IP do Arduino, este irá gerar uma página em HTML em seu programa para ser enviada como resposta ao computador. Quando a temperatura limite configurada na programação for atingida, uma função “envioemail” chama o programa “GoBetWino” que chama a página “localhost/sendmail.php” que enviará o e-mail.

A Figura 7 apresenta um diagrama de fluxo de dados (DFD) simplificado para mostrar os fluxos de dados entre os componentes do sistema proposto.

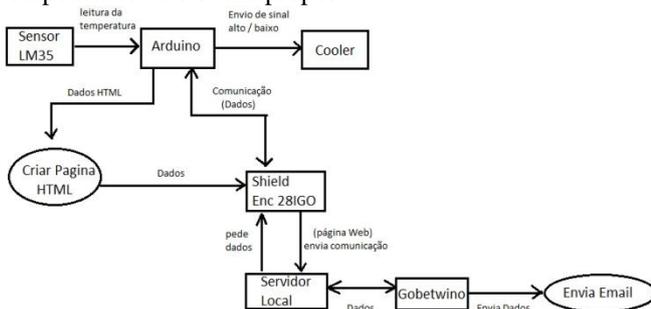


Fig. 7. Representação gráfica simples de DFD

A Figura 8 mostra uma imagem da página em HTML criada para informar a temperatura atual do ambiente e a ação a ser tomada pelo usuário.



Fig. 8. Página do HTTP criada

1) Wamp

O Wamp é uma aplicação que conjuga várias tecnologias oferecendo ferramentas como Apache, MySQL e PHP para Windows. Através do *wampserver* consegue-se um ambiente de desenvolvimento Web, permitindo a criação de aplicações Web com Apache, banco de dados MySQL e PHP. A escolha do Wamp foi devida à facilidade de manuseio e por ser completo.

Neste projeto, o Wamp cumpre o papel principal de ser o servidor central responsável por enviar mensagens através de e-mail com a criação do arquivo “sendmail.php”.

Após a configuração do arquivo “sendmail.php”, é necessário também configurar o arquivo “sendmail.exe” que fica dentro do diretório principal do Wamp, chamado “sendmail.ini”. É muito importante a configuração do “sendmail.exe” pois sem ela não é possível enviar o e-mail através do “sendmail.php”.

Ao configurar a porta de envio SMTP, a linha “smtp_ssl = none” indica que não será usada a criptografia para envio das informações de usuário e senha. A linha “default_domain = eel.ufu.br” indica que usaremos o domínio padrão eel.ufu.br.

Como o trabalho precisava ser apresentado na universidade, houve a necessidade de contatar o centro de tecnologia da informação (CTI) para saber sobre o sistema de proteção da rede dentro da instituição. Pela versatilidade e robustez do “sendmail” que possui um “auth username” e um “auth password”, ou seja, uma comunicação autenticada necessária de e-mail, foi possível realizar a comunicação dentro da instituição.

Além disso, foi necessário configurar o arquivo “php.ini” da seguinte forma:

- A linha “SMTP = smtp.eel.ufu.br” indica qual o servidor SMTP será utilizado no envio do e-mail. Neste trabalho, utilizou-se o servidor SMTP da Faculdade de Engenharia Elétrica da Universidade Federal de Uberlândia;
- A linha “smtp_port = 25” configura a porta 25 para envio do e-mail;
- A linha “sendmail_path = "D:\wamp\sendmail\sendmail.exe -t”” indica o diretório no qual se encontra o arquivo “sendmail.exe” responsável pelo envio do e-mail.

A Figura 9 mostra a imagem da página notificando que o e-mail foi enviado com sucesso através do servidor de e-mail.

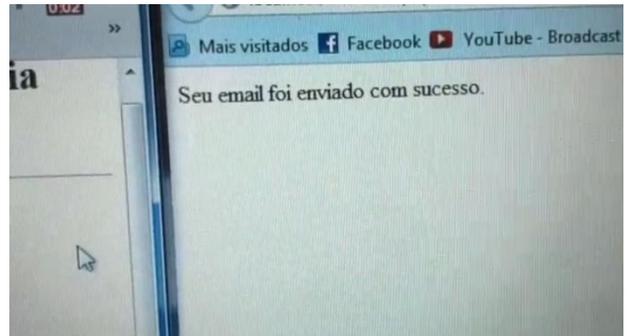


Fig. 9. Visualização da notificação do servidor e e-mail

A Figura 10 mostra a imagem da caixa de entrada do *browser* ilustrando o recebimento do e-mail enviado pelo sistema proposto. Na Figura 11, verifica-se o conteúdo do e-mail recebido pelo usuário, com o alerta sobre a temperatura corrente e as providências a serem tomadas.

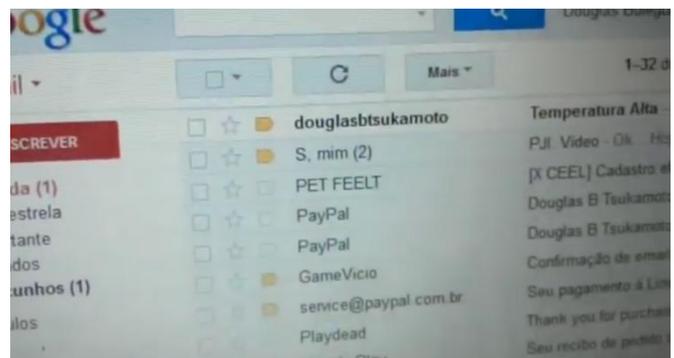


Fig. 10. Caixa de entrada de e-mail

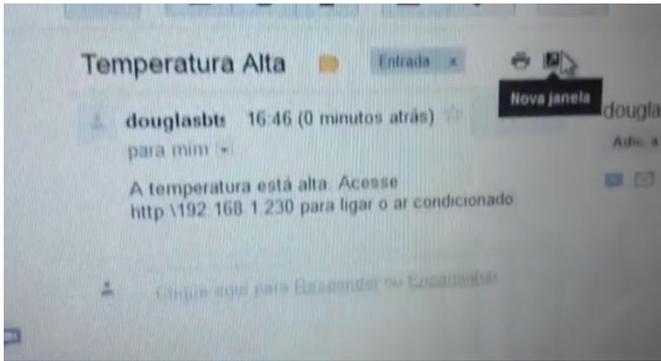


Fig. 11. E-mail enviado

2) Stunnel

O *Stunnel* [10] é um programa *open source* multi-plataforma de computadores, usado para fornecer serviço de tunelamento TLS (segurança da camada de transporte) /SSL (protocolo de camada de *sockets* segura). Para redes que não possuem seu próprio servidor SMTP, ele é necessário para o envio de e-mails, pois é o responsável por configurar o protocolo SSL responsável pela comunicação da rede local com o servidor SMTP externo. Neste trabalho, o *Stunnel* foi utilizado para configurar uma comunicação SMTP com o servidor do gmail.

3) Comunicação via Internet

O sinal começa no sensor de temperatura, que é enviado para a porta analógica 0 do Arduino. Posteriormente, o valor da temperatura é calculado e convertido para *string* para que seja possível a impressão na tela. O valor da temperatura então é enviado para a impressão na página em HTML através do intermediário Ethernet Shield ENC28j60. Para o envio do e-mail, há um comando “if” no código do programa para que quando a temperatura atingir um certo limite, o programa “GoBetWino” abra a página “localhost/sendmail.php” para esse envio.

Para que possa ser feita a conexão via internet é necessário realizar as configurações descritas nesta seção, além de saber o IP do roteador, configurando também o roteador para que seja aberta uma porta somente para o Arduino. Neste trabalho foi utilizado o cabo RJ45 para fazer a comunicação entre o Arduino e o roteador. Acredita-se que se for feita uma conexão via Ethernet Shield com *wireless*, será possível a conexão sem fio.

IV. IMPLEMENTAÇÃO

Nesta seção estão apresentados os fragmentos de programas utilizados na configuração e implementação do *software* responsável pela comunicação via Internet entre o usuário e o sistema proposto.

A. Esquema do software do sistema proposto

A Figura 12 apresenta um esquema do sistema proposto destacando-se a parte do *software* com as suas funcionalidades.

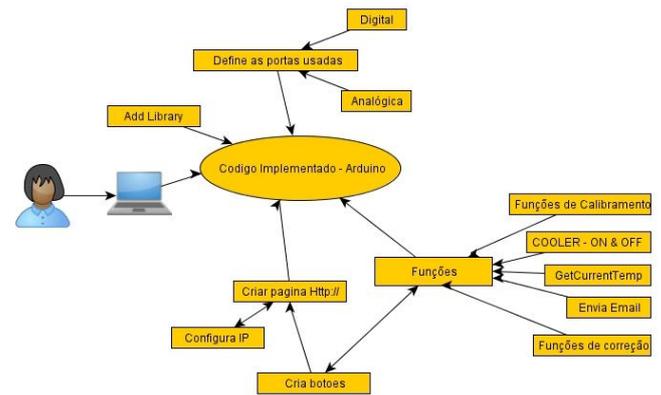


Fig. 12. Esquema geral do sistema proposto com destaque para o *software*

B. Função Sendmail.php

Como descrita na seção anterior, segue abaixo um exemplo de codificação da função “sendmail.php”.

```
<?php
    $email='douglasbtsukamoto@gmail.com';
    $resultado = mail($email,
    'Temperatura Alta', 'A temperatura está
    alta. Acesse http:\\192.168.1.230 para
    ligar o ar condicionado.');
```

```
    if($resultado)
    {
        echo 'Seu email foi enviado com
    sucesso.';
    }
    else
    {
        echo 'Não foi possível enviar seu
    email.';
    }
}
```

A função “sendmail.php” é a responsável por enviar o e-mail ao usuário. Ela fica armazenada no banco de dados do servidor criado pelo *wampserver* e será chamada pelo código criado dentro do Arduino.

C. Configuração Sendmail.ini

No Wamp é necessário configurar o arquivo “sendmail.exe” que fica dentro do diretório principal chamado “sendmail.ini”. Tal configuração está exemplificada a seguir:

```
[sendmail]
smtp_server=smtp.eel.ufu.br
smtp_port=25
smtp_ssl=none
error_logfile=error.log
auth_username=douglasbtsukamoto
@eel.ufu.br
auth_password=*****
pop3_server=-----
-----
pop3_username=
pop3_password=
```

```
force_sender=douglasbtsukamoto@eel.ufu.br
force_recipient=
hostname=
```

D. Configuração php.ini

O código abaixo exemplifica a configuração do arquivo “php.ini” para que o envio de e-mail fosse possível com a utilização do servidor SMTP da Faculdade de Engenharia Elétrica da Universidade Federal de Uberlândia.

```
[mail function]
SMTP = smtp.eel.ufu.br
smtp_port = 25
sendmail_path = "C:\wamp\sendmail\
                sendmail.exe -t"
```

E. Configuração do Stunnel

Finalmente, a codificação a seguir ilustra a configuração do *Stunnel* de uma comunicação SMTP com o servidor do gmail.

```
[smtpg]
accept = 127.0.0.1:25
connect = smtp.gmail.com:465
```

V. CONCLUSÃO

Através deste trabalho verificou-se o quanto a plataforma baseada no dispositivo Arduino é uma ferramenta versátil, de fácil implementação e manuseio para os propósitos do desenvolvimento do projeto proposto. A plataforma mostrou-se versátil, de baixo custo e suficiente para se atingir os objetivos propostos neste projeto.

Com o uso conjunto do Arduino Uno com o módulo Ethernet Shield, foi possível a conexão do dispositivo à Internet de forma que o usuário possa monitorar remotamente a temperatura de um ambiente que está localizado a uma longa distância. O sistema desenvolvido em questão é capaz de emitir mensagens de alerta ao usuário em caso de excesso de temperatura e receber comandos desse usuário para corrigir a temperatura desse ambiente, através da Internet.

O produto final do trabalho, ou seja, o sistema integrado de *hardware* com *software* que viabiliza a comunicação com

o usuário via Internet e o controle de equipamento pelo usuário localizado a longas distâncias foi testado e o seu funcionamento devidamente comprovado.

Como proposta para trabalhos futuros, tem-se a alternativa de substituir os cabos de conexão por dispositivos Bluetooth e a adaptação para o uso de *smartphones*, aumentando-se a gama de utilização do sistema.

Conclui-se ainda que o projeto pode se adequar tanto aos segmentos comerciais e industriais quanto para o uso doméstico.

REFERÊNCIAS

- [1] ALVES, L. J. *Instrumentação, Controle e Automação de Processos*, 2ª Ed., LTC, 2010.
- [2] TIMMIS, H. *Practical Arduino Engineering*, 1ª Ed., Technology in Action, 2011.
- [3] ARDUINO. Disponível em: <<http://www.arduino.cc/>>. Acesso em: 20 julho 2012.
- [4] LIQUIDWARE. *Arduino Uno*. Disponível em: <<http://www.liquidware.com/shop/show/ARD-UNO>>. Acesso em: 20 agosto 2012.
- [5] PRECIOLANDIA. *Ethernet Shield ENC28J60*. Disponível em: <<http://www.preciolandia.com/br/ethernet-shield-enc28j60-codigo-arduino-83wmd5-a.html>>. Acesso em: 11 julho 2012.
- [6] BEBEDOR ELETRONICO. *Sensor de Temperatura LM35*. Disponível em: <<http://bebedoreletronicoprojeto.webnode.com.br/news/sensor-de-temperatura-lm35/>>. Acesso em: 20 agosto 2012.
- [7] HARDWARE SECRETS. *How to Build a Web Server with PHP 5 and MySQL 5 Support*. Disponível em: <<http://www.hardwaresecrets.com/article/How-to-Build-a-Web-Server-with-PHP-5-and-MySQL-5-Support/436>>. Acesso em: 10 agosto 2012.
- [8] PHP FREAKS. *How to Setup Sendmail.ini for PHP with hMail Server?*. Disponível em: <<http://forums.phpfreaks.com/topic/226774-how-to-setup-sendmailini-for-php-with-hmail-server/>>. Acesso em: 10 agosto 2012.
- [9] RAEL. *Enviando emails com PHP no Windows*. Disponível em: <<http://raelcunha.com/sendmail.php>>. Acesso em: 10 agosto 2012.
- [10] STUNNEL. *Stunnel: Home*. Disponível em: <<https://www.stunnel.org/index.html>>. Acesso em: 28 julho 2012.