Uma arquitetura para Sensoriamento e Tratamento de Eventos voltada à Área de Segurança para Controle e Rastreio de Usuários em Ambientes Físicos

Walther F. Pedrozo[1], Alexandre Silva Rodrigues[2], Bruno S. Alves[2],Clóvisson L. Rosa[1],Tiago A. Rizzetti[1]

[1]Colégio Técnico Industrial de Santa Maria (CTISM) – Universidade Federal de Santa Maria (UFSM), CEP– 97.105-900 – Santa Maria – RS – Brasil

[2]Centro de Tecnologia (CT) – Universidade Federal de Santa Maria (UFSM), CEP-97.105-900 – Santa Maria – RS – Brasil

{waltherpedrozo,clovisson.lopes,alexandre.rodrigues}@redes.ufsm.br, rizzeti@gmail.com,bdalves@inf.ufsm.br

Abstract. *Physical environments of educational institutions and workplaces are increasingly equipped with high quality materials and with high financial value, requiring entry control only from authorized persons. This architecture proposes to solve this problem by implementing a system of control and tracking users in physical environments, which facilitates those responsible for managing the environments to manage the access of persons with authorization.*

Resumo. *Ambientes físicos tanto de instituições de ensino como de locais, trabalhos estão cada vez mais equipados com materiais de alta qualidade e com alto valor financeiro, sendo necessário o controle de entrada somente de pessoas autorizadas. Esta arquitetura se propõem a solucionar este problema implementando um sistema de controle e rastreio de usuários em ambientes físicos, que facilite aos responsáveis pela administração dos ambientes gerenciarem o acesso de pessoas com autorização.*

1. Contextualização

Na maioria das instituições existem ambientes físicos onde se deseja restringir o acesso somente para pessoas autorizadas. O sistema de chaves físicas apresenta falhas de segurança, pois não possui, vinculado a si próprio, dispositivos que limitem o acesso de um usuário não autorizado, nem auditoria de uma forma eficiente.

Para impossibilitar o acesso a de pessoas indevidas a esses ambientes, torna-se necessário um método que permita o acesso somente para pessoas cadastradas, registrando inclusive as tentativas de acesso. É desejado que o método de acesso possua mecanismos de registro e controle, preenchendo as prerrogativas de protocolos da família AAA, autenticação, autorização e auditoria [Santos, 2007].

Outro fator importante é a centralização das informações do sistema. Dessa forma, é possível solucionar o problema das fechaduras eletrônicas stand, mantendo um controle sobre diversos dispositivos e ambientes, utilizando uma base central para autenticação, autorização e auditoria dos dispositivos do sistema. Com base nisso, este trabalho apresenta aprimoramentos no controle de acesso implementado na plataforma

ESC (*Environment Security Control*) que permite controle de acesso a ambientes, gerenciando permissões e auditoria, baseado na interação entre vários dispositivos físicos e um software gerente. Esse, por sua vez, caracteriza-se como um middleware para tratamento dos eventos gerados pelos diversos dispositivos integrados nesta plataforma. A principal mudança em relação a anterior foi a substituição da plataforma Arduino, que apresentou problemas de instabilidade e baixa capacidade processamento, pelo Raspberry Pi, o qual soluciona esses problemas apresentados em função da utilização de um hardware com maior capacidade computacional.

2. Trabalhos Relacionados

O controle de acesso apresenta uma série de problemas a serem solucionados. Nesse contexto, é possível encontrar diversas propostas que visam solucionar problemas semelhantes, utilizando diferentes abordagens. Entre as soluções acadêmicas, podemos destacar:

Sentinel: um engenho Java para controle de acesso RBAC, este trabalho descreve um controle de acesso baseado em papéis (RBAC), que pode atuar de forma genérica em diferentes tipos de aplicações. A autorização baseia-se na atribuição de diferentes tipos de privilégios para cada grupo de usuários. Embora exista um módulo de auditória, esse não é muito desenvolvido, visto que, a principal preocupação dessa proposta é realizar o processo de autenticação. [Mattos, 2003].

Sistema de Controle de Acesso Utilizando Dispositivos Embarcados, apresenta uma proposta para o controle de acesso às salas de aula de uma universidade. Esse sistema é constituído por um middleware de um servidor de autenticação. O middleware é responsável por realizar a autenticação de usuários por meio da leitura de tags RFID e enviar para o servidor de autenticação, utilizando a rede *ethernet*. O servidor de autenticação atua de forma centralizada, atendendo as requisições de todos os middlewares. O autor ressalta a necessidade de uma interface amigável para facilitar a administração do sistema, embora não esteja implementada [Peixoto, 2013].

A Arquitetura ESC diferencia-se das citadas anteriormente, por realizar o registro de todos os eventos tratados, possui uma interface de interação com o sistema e atua de forma centralizada sobre dispositivos físicos que integram o sistema.

Além disso, existem soluções proprietárias para resolver estes problemas. Entre elas, podemos citar:

Controle de Acesso Remoto Siemens: permite cadastrar usuários, agendar horários para cada usuário e possui características de auditoria, informando algum acesso em horário indevido. Além disso, é possível vincular esse sistema a um sistema de alarme monitorado 24 horas [Heinsch, 2011].

Controle de acesso NibTec: possibilita que diferentes dispositivos de entrada sejam utilizados, como por exemplo: teclados numéricos, cartões com tecnologia de radiofrequência (RFID) e dispositivos biométricos. Além disso, permite realizar auditorias, gerando relatórios sobre entrada e saídas de usuários [Heinsch, 2011].

Por tratar-se de soluções proprietárias, as empresas não disponibilizam maiores informações sobre os códigos-fonte e apresentam um custo maior para serem adquiridos pelo usuário final. Nesse aspecto a Arquitetura ESC se destaca, utilizando plataformas

abertas para desenvolver os *hardwares* e *softwares*, permitindo alterações e adaptações para a utilização em diferentes ambientes.

3. Arquitetura ESC

A arquitetura ESC foi projetada com vistas a proporcionar uma plataforma para monitoramento de ambientes físicos, realizando tratamento de eventos via sensores e atuação no ambiente através de atuadores. É portanto um sistema de tratamento de eventos onde através de informações lógicas controla-se parâmetros do ambiente físico.

A arquitetura ESC é implementada sob 3 diferentes componentes, o ESCMA (ESC *Manager*), ESCHA (ESC *Hardware*) e ESCI. (ESC *Interface*) Na figura 1 são apresentados esses componentes e a relação existente entre eles.

O ESCHA é um conjunto de sensores e atuadores conectados a um microcontrolador , com capacidade de reconhecer eventos e transmitir e/ou receber informações através de uma rede de comunicação baseada em protocolo IP. O ESCMA é um software de gerenciamento que atua de forma centralizada, sendo responsável pela comunicação com todos os dispositivos que compõem o sistema, realizando a coleta de eventos e seu respectivo tratamento. O ESCI realiza a interação entre o gerente (ESCMA) e o usuário do sistema, podendo configurar módulos, conexões e o gerenciamento das conexões com o banco de dados.

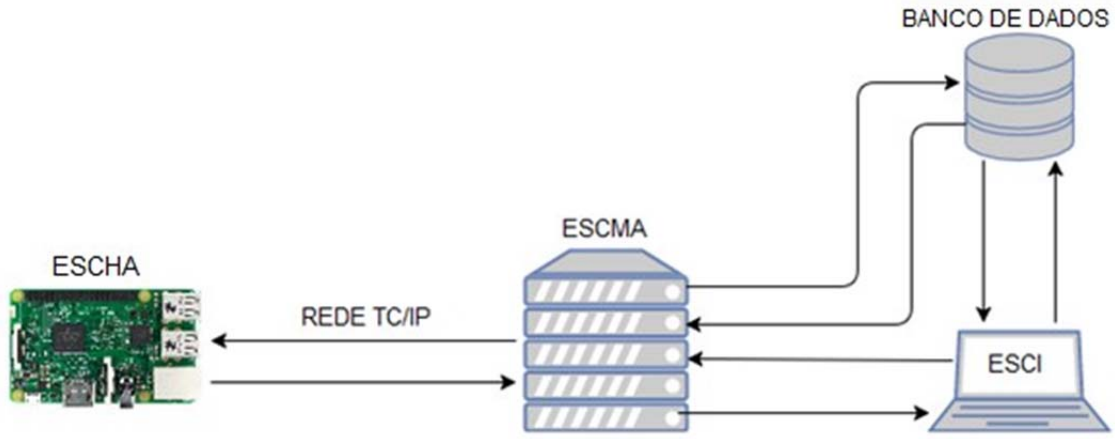

Figura 1. Arquitetura ESC

4. Comunicação

A comunicação entre o ESCHA e o ESCMA utiliza a pilha de protocolos TCP/IP. Esta oferece as funcionalidades necessárias para o envio de dados, eliminando a necessidade de uma nova rede específica.

Além disso, é necessário um protocolo de comunicação para enviar e receber mensagens, implementado em ambos os subsistemas, de maneira que seja possível o entendimento de uma informação enviada. Dessa forma, vários critérios devem ser observados, como por exemplo: a troca de mensagens deve ser sincronizada, necessidade de uma forma de criptografar as mensagens, possibilitar que qualquer dispositivo possa ser utilizado como sensor ou atuador e respeitar as limitações de processamento do hardware desenvolvido.

O protocolo desenvolvido baseia-se no envio de uma string, conforme apresentada na Figura 2. Essa string é montada conforme um padrão adotado nos subsistemas. Para realizar a comunicação é utilizado o *modelo cliente-servidor*, nesse cenário o ESCHA atua como o cliente, que envia uma solicitação ao servidor (ESCMA), que realiza o processamento dos dados e envia uma resposta. Visando manter atualizado na interface o estado do gerente e dos dispositivos, a arquitetura utiliza o mecanismo de *pooling,* no qual é feito o envio de uma mensagem para o ESCMA informando o estado do dispositivo, caso não haja geração de eventos em um intervalo de tempo preestabelecido.

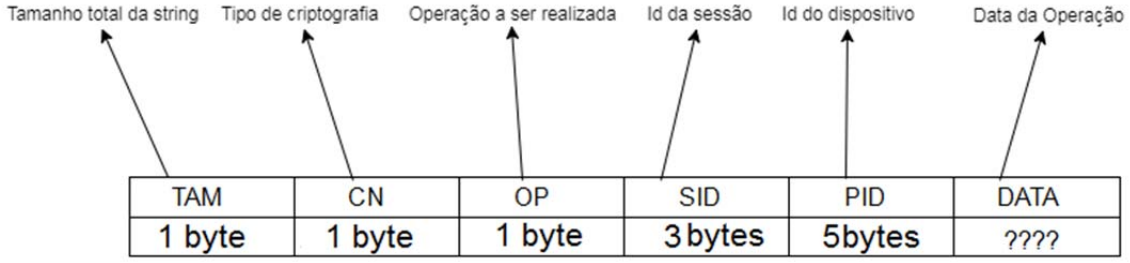

Figura 2. Pacote padrão

4.1. ESCHA

Os primeiros componentes utilizados como dispositivo de hardware na arquitetura ESC foi a plataforma arduino. Porém, através do intenso uso experimentado no estudo de caso de sua implantação, verificou-se que a baixa capacidade computacional gera eventualmente instabilidades na comunicação causando problemas de difícil rastreamento e replicação. Desta forma, optou-se pela experimentação de plataforma de hardware mais robustas, capazes de realizar operações de criptografia e comunicação de rede sem sobrecarga ao sistema, a solução natural foi o emprego da plataforma Raspberry PI 3. A principal contribuição dessa plataforma é facilita a integração entre os dispositivos eletrônicos (sensores e atuadores) a comunicação de rede através da implementação tradicional da pilha de protocolos *TCP/IP* utilizadas por sistemas operacionais de propósito geral, como o Linux.

Uma grande vantagem no uso da arquitetura baseada em Raspberry é que se pode utilizar um sistema de propósito geral, baseado em Linux, capaz de oferecer interfaces de programação e utilização idênticas àquelas encontradas em computadores tradicionais. Isso facilita, além do desenvolvimento em plataformas *cross compile* de fácil simulação, a possibilidade rastreabilidade de bugs de forma mais fácil e eficiente daquela presente na plataforma Arduino (somente informações via serial e, que causam impacto significativo de recursos. Além disso, ela apresenta diversas vantagens em relação a outras plataformas, como processamento, software *open source* e a possibilidade de expansão utilizando as portas GPIO. Na versão dois da arquitetura foi desenvolvida uma placa, que conta com: módulo e antena para leitura do cartão RFID, componentes eletrônicos necessários ao acionamento de uma fechadura eletromagnética, além de Leds indicadores do funcionamento da mesma.

O ESCHA, basicamente pode ser visto como um conjunto de sensores e atuadores, que realiza uma comunicação com um gerente centralizado (ESCMA), por meio de redes TCP/IP.

A arquitetura ESC, por ser escalável torna-se possível a adição de dispositivos físicos sem necessitar alteração no gerente (ESCMA), tornando o sistema mais flexível a mudanças.

4.2 ESCMA

Esse subsistema trata do módulo de gerenciamento centralizado, projetado para armazenar as configurações de todo sistema, provendo escalabilidade, flexibilidade, e facilidade de uso. Portanto estas características permitem constantes modificações e adição de novas categorias de dispositivos físicos, através da adição de plugins.

Nesse contexto, o ESCMA pode ser dividido em módulos: dispositivo físico, gerente, base de dados, auditoria e interface. Desta forma, cada módulo exerce uma atividade especifica, conforme pode ser observado na figura 3.

A conexão com o banco de dados acontece, quando o ESCMA recebe um evento e precisa consultar se o usuário tem permissão de acesso ao ambiente desejado, retornando ao dispositivo físico a resposta. Ela também é utilizada para armazenar registros de acessos permitidos e não permitidos, para utilização em caso de auditoria.

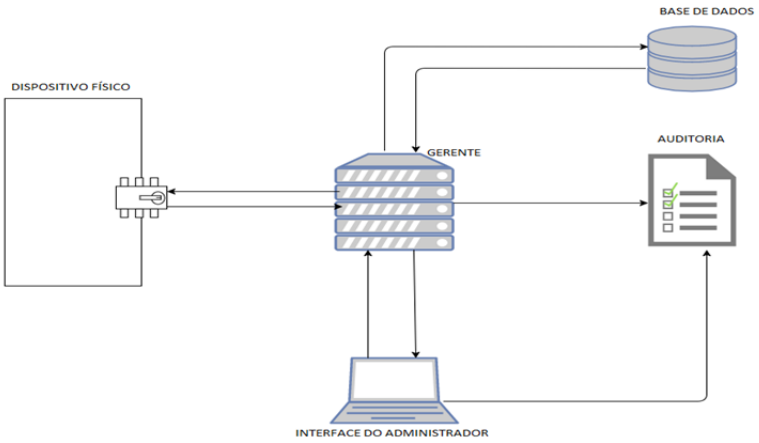

Figura 3. Conexão entre módulos

Na versão um da Arquitetura ESC, a coleta de informação utilizava o método *polling* o qual buscava informações em um intervalo de tempo definido, essa técnica era utilizada devido às limitações do hardware dessa versão, pois impediam que esse funcionasse como um cliente, apresentando limitações no buffer. Visando evitar o desperdício de recursos computacionais a versão dois da arquitetura ESC, modificou o método de busca de informação, agora os dispositivos físicos iniciam comunicação com o gerente somente quando houver alguma alteração no ambiente, sendo assim só haverá comunicação quando um evento precise ser tratado pelo gerente, esse modelo é conhecido como *modelo cliente-servidor*, no qual os dispositivos (ESCHA) são os clientes e o gerente (ESCMA) atua como servidor que espera a comunicação dos clientes.

4.3 ESCI

Com a necessidade de haver comunicação entre o administrador e gerente, foi projetada uma interface web, utilizando a linguagem de programação *PHP* e a linguagem de marcação *HTML* e *CSS*. Essa interface pode coordenar a configuração de todo sistema, podendo interagir. A principal vantagem da interface web é a flexibilidade que a mesma oferece, pois é acessível de qualquer lugar, desde que o usuário possua cadastro e permissões de acesso à página. O ESCI oferece diversas funções aos usuários como adicionar e remover dispositivos de entrada e usuários, verificar o status do gerente, ver os dispositivos existentes, realizar auditorias, gerar relatórios de acesso e tentativas de acesso tanto por dispositivos quanto por usuários.

5. Resultados e Conclusões

Visando analisar o desempenho e confiabilidade da arquitetura proposta, foram feito dois testes: Um utilizando o Arduino e outro à plataforma Raspberry.

Os testes foram realizados em duas sessões de 3 minutos cada. Durante cada uma foram enviadas 10 tags com autorização e 10 sem. Nos testes foram enviadas 40 solicitações de acionamento de dispositivo ao gerente.

Nos testes com o Arduino foi utilizado o cenário onde é feita a passagem de dois cartões RFID nos dispositivos, um com permissão de entrada e outro sem.

Para calcular o tempo decorrido entre a solicitação de abertura e a resposta do gerente, foi feita uma subtração do tempo inicial da solicitação com o de resposta do gerente. Após foi feito a média desses valores, retornando como resultado 1,4784 segundos para o cartão autorizado e 1,2955 segundos para o não autorizado.

No raspberry foram enviadas 40 tags diretamente da plataforma, sendo 20 com autorização e 20 sem. Após houve a adição de uma função no código onde retorna o tempo de resposta do gerente por tentativa e uma média de todas as tentativas. O tempo médio de resposta para tags com permissão foi de 0,19488 segundo e para tags sem permissão de 0,1846265 segundo, tendo uma pequena variação entre os tempos de resposta.

Analisando o Gráfico 1 e 2, é possível perceber que existe uma grande diferença de tempo de resposta na plataforma Arduino e Raspberry, utilizando as tags válidas a diferença foi de 1,283592 segundos. Já em tags inválidas foi de 1,0958735 segundos.

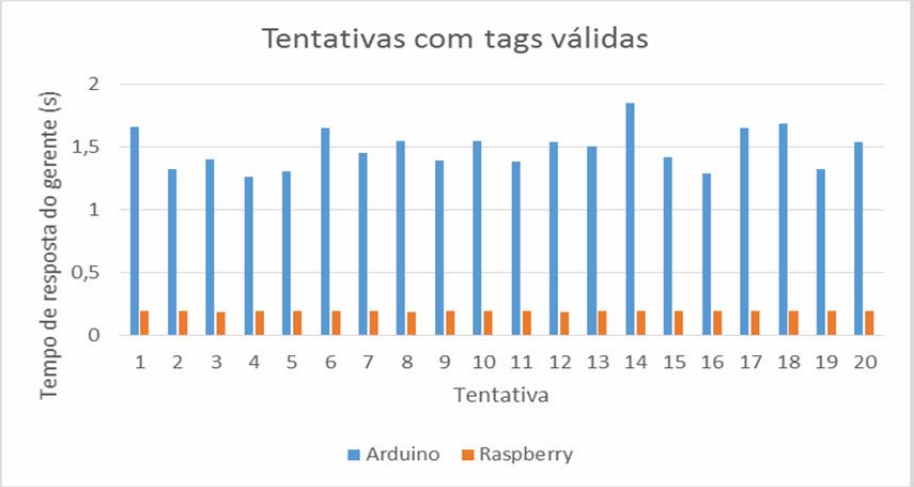

Gráfico 1. Comparação do tempo de resposta do gerente para tags válidas entre as duas plataformas.

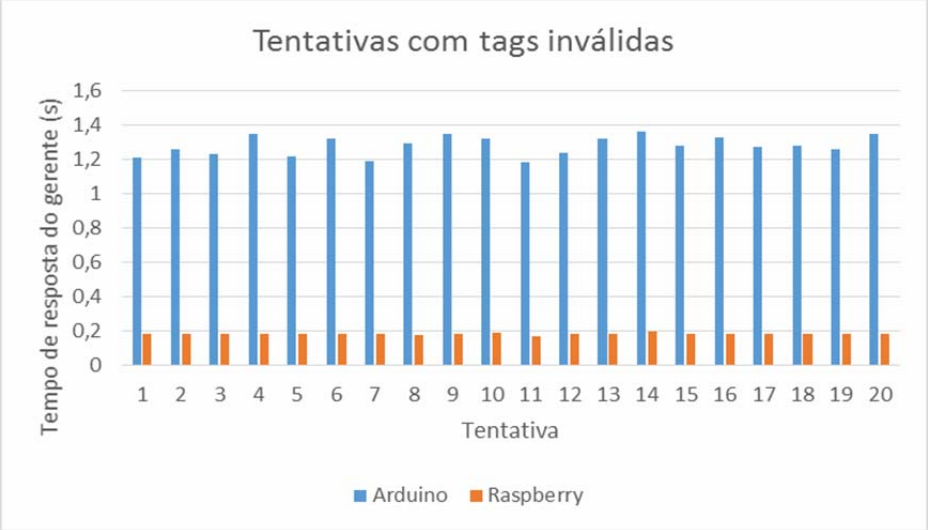

Gráfico 2. Comparação do tempo de resposta do gerente para tags inválidas entre as duas plataformas.

A partir desses testes é possível observar a estabilidade do gerente, pois mesmo tendo sido utilizado diversas vezes em um curto espaço de tempo, não houve problemas em seu sistema, o que comprova o quão escalável e confiável a arquitetura ESC é.

Outro ponto a ser observado é a diferença nos tempos de resposta utilizando as duas plataformas, constata o quão relevante foi à substituição da plataforma Arduino pelo Raspberry, o principal ponto para a diferença no tempo de respostas está no tempo de criptografia, pois o raspberry por apresentar melhores componentes computacionais, sendo assim o tratamento das mensagens é feito de forma mais rápida.

5. Trabalhos Futuros

A partir de agora pretende-se adicionar mais funcionalidades a arquitetura ESC, visto que a mesma já se mostrou bastante flexível. Um dos exemplos de funcionalidades a ser incrementada é o controle da temperatura e umidade do ambiente onde a placa atua, visando evitar danos físicos as placas.

Referências

Gilmore, W. J., (2010), Beginning PHP and MySQL.

Heinsch, L. R. (2011), Sistema automatizado para controle de acesso ás salas do CTISM. Trabalho de Conclusão de Curso, Universidade Federal de Santa Maria.

Hailperin, M. (2007), Operating systems and middleware: interaction.

Mattos, C. L. A. (2003), Sentinel: um engenho Java para controle de acesso RBAC. Trabalho de Conclusão de Curso, Universidade Federal de Pernambuco.

Mendes, Antonio. (2002), Arquitetura de Software: desenvolvimento orientado para arquitetura.

McRoberts, M. (2015), Arduino Básico – 2º Edição.

Monk, Simon. (2016), Movimento, Luz e Som com Arduino e Raspberry; Tradução

Peixoto, T. M. (2013), Sistema de Controle de Acesso Utilizando Dispositivos Embarcados. Trabalho de Conclusão de Curso, Universidade Federal de Juíz de Fora.

Quintas, D. L. (2012), Controle de Acesso Utilizando uma Rede de Microcontroladores. VII Jornada de Iniciação Científica, Desenvolvimento Tecnológico e Inovação do Ifes.

Raguzzoni, J. C. M., Heinsch L. R. e Rizzetti, T. A. (2012), Uma arquitetura para desenvolvimento de dispositivos de autenticação e acesso a espaços físicos.

Robertson Craig, Ronald C. Beavis; TANDEM: matching proteins with tandem mass spectra. Bioinformatics 2004; 20 (9): 1466-1467. doi: 10.1093/bioinformatics/bt092.

Santos, A. (2007), Gerenciamento de Identidades.

Silva, Maurício Samy (2011), CSS3: desenvolva aplicações web profissionais com o uso dos poderosos recursos de estilização da CSS3.

Sugimoto, G.M., Aguiar, L.K., (2011), Implementação de Protocolos da Pilha TCP/IP.