UNIVERSIDADE FEDERAL DE SANTA MARIA
CENTRO DE TECNOLOGIA
CURSO DE CIÊNCIA DA COMPUTAÇÃO

Sistema automatizado para controle de acesso às salas do CTISM

Lamarck Ribas Heinsch

Santa Maria, RS, Brasil.

2011

Sistema automatizado para controle de acesso às salas do CTISM

Lamarck Ribas Heinsch

Trabalho de Conclusão de Curso apresentado no Curso de Ciência da Computação do Centro de Tecnologia da Universidade Federal de Santa Maria como requisito parcial para obtenção do grau de
Bacharel em Ciência da Computação

Orientador: Profª. Drª. Márcia Pasin

Santa Maria, RS, Brasil.

2011

Universidade Federal de Santa Maria

Centro de Tecnologia
Curso Graduação em Ciência da Computação

A Banca Examinadora, abaixo assinada, aprova o
Trabalho de Conclusão de Curso

Sistema automatizado para controle de acesso às salas do CTISM

elaborado por
Lamarck Ribas Heinsch

como requisito final para obtenção do grau de
Bacharel em Ciência da Computação

BANCA EXAMINADORA:

_____

Profª. Drª. Márcia Pasin (UFSM)
(Presidente / Orientadora)

_____

Prof. Dr. Benhur de Oliveira Stein (UFSM)

_____

Prof. Msc. Tiago Antônio Rizzetti (UFSM)

Santa Maria, 19 de dezembro de 2011.

AGRADECIMENTOS

Esta é a ultima parte que eu escrevo deste trabalho, acredito que muitas pessoas sentem saudades do trabalho assim que ele termina. Começou com uma ideia de criar uma coisa simples no local de trabalho que virou um monstro a medida que o tempo foi passando. Logo depois consegui, para minha sorte, que a minha orientadora Márcia (abraço para ela), aceitasse meu projeto para que fosse utilizado como trabalho de graduação.

Agradeço aos grandes amigos do SSI, Tiago, Jeann, Dalvane por estarem lá e ajudarem em vários momentos deste projeto, tanto na parte do desenvolvimento dos dispositivos, quanto nas bobagens entre uma Coca Cola e outra.

Obviamente que vou agradecer à minha namorada Caroline, por me puxar a orelha e me mandar fazer o trabalho, e deixar de ser preguiçoso guri, mas isso muito me impulsionou para frente.

Finalmente agradeço à minha família, que sempre esteve por perto, seja lá o que isso signifique. Pai, avós, Carol, e galera do SSI, Aquele Abraço.

E não importa o que venha pela frente, pois essa etapa foi só o começo.

RESUMO

Trabalho Final de Curso
Curso de Graduação em Ciência da Computação - Bacharelado
Universidade Federal de Santa Maria

Sistema automatizado para controle de acesso às salas do CTISM

Autor: Lamarck Ribas Heinsch
Orientadora: Márcia Pasin
Local e Data da Defesa: Santa Maria, 19 de dezembro de 2011.

Controle de acesso é uma característica importante para a segurança de muitos sistemas e é interessante dispor de soluções automatizadas principalmente em locais onde circulam grande quantidade de pessoas. Este trabalho consiste na elaboração de um sistema para automatizar e gerar auditoria no controle de acesso a locais restritos do Colégio Técnico Industrial de Santa Maria.
No escopo deste trabalho, o objetivo é projetar e desenvolver a aplicação gerente, responsável por tratar as requisições de acesso. Após a entrada dos dados, o sistema verifica em uma base de dados as permissões daquele usuário a partir das informações recebidas. Com esse conjunto de informações é possível gerar a auditoria desejada para o controle dos ambientes restritos. A resposta à requisição do usuário é enviada ao dispositivo controlado, o qual é responsável pelo acionamento de um atuador, como por exemplo, abrir uma porta. Além disso, consta nesse trabalho uma comparação entre soluções comerciais e soluções acadêmicas para problemas relacionados e uma revisão breve dos principais pontos que constituem a segurança da informação. Finalmente, são apresentados a metodologia e os resultados dos testes do sistema em cenário real e as considerações finais a respeito desse trabalho.

Palavras-chave: Auditoria, Informação, Controle de acesso.

ABSTRACT

Trabalho Final de Curso
Curso de Graduação em Ciência da Computação - Bacharelado
Universidade Federal de Santa Maria

Automated system for control access to the CTISM's rooms

Author: Lamarck Ribas Heinsch
Advisor: Márcia Pasin
Defense Place and Date: Santa Maria, December 19, 2011.

Access control is an important issue to allow safety in many systems, and is interesting to have automated solutions, especially in places where moving large number of people. This work presents a solution involving low-cost hardware and software to automate and generate audit to access control for restricted areas. The Colégio Técnico Industrial de Santa Maria of UFSM is used as scenario.
In this project, the goal is to design and develop the manager software, responsible for handling requests for access. When the user inserts his credentials, the system checks in a database the permissions matching between received data and data previously stored. With such information, it is possible to generate the desired control audit of staff. The answers to the user request is sent to the controlled device, which is responsible for driving an actuator,, such as opening a door. In addition, this work presents a comparison involving mechanisms provided by current commercial and academic solutions and a review focusing on the main points of information security. Finally, this work presents the methodology, test results on the real scenario and concluding remarks.

Keywords: Audit, Information, Access control.

LISTA DE ABREVIATURAS E SIGLAS

CTISM	Colégio Técnico Industrial de Santa Maria
DAC	*Discretionary Access Control*
HTTP	*Hypertext Transfer Protocol*
JVM	*Java Virtual Machine*
LDAP	*Lightweight Directory Access Protocol*
MAC	*Mandatory Access Control*
MD5	*Message Digest algorithm 5*
PIC	*Programming Interface Controller*
RADIUS	*Remote Authentication Dial In User Service*
RBAC	*Role Based Access Control*
RFID	*Radio-Frequency Identification*

SUMÁRIO

1 APRESENTAÇÃO

Em grande parte das instituições a segurança é um fator imprescindível, pois nestes locais existe um maior controle de fluxo de pessoas a pontos reservados, bem como restrições de acesso devido à existência de objetos de valor (informações, serviços, bens materiais como joias, carros, etc.).

Nestes setores é necessário um cuidado maior referente a quem pode ou não ter acesso ao ambiente por questões de segurança, como por exemplo, a restrição de entrada em uma sala com dezenas de computadores que mantêm os serviços necessários para a estrutura computacional funcionar corretamente, um cofre que mantém objetos de valor, etc. Nesses ambientes, o fluxo de funcionários e outras pessoas deve ser restrito, porém nem sempre é assim. Aliado a esta problemática, muitas vezes precisamos de soluções envolvendo baixo custo de implantação e manutenção, como é o caso de ambientes acadêmicos. Neste cenário, destaca-se o Colégio Técnico Industrial de Santa Maria (CTISM), onde o acesso dos funcionários a ambientes não possui um modo de controle eficiente de entrada e saída, sendo realizado manualmente por um funcionário através de um caderno de registro de empréstimo de chaves desses locais.

1.1 Cenário atual do CTISM

O CTISM, Colégio Técnico Industrial de Santa Maria, é uma escola de ensino médio e profissionalizante, vinculada à UFSM. Neste ambiente circulam livremente pelos corredores mais de 500 pessoas, divididos entre alunos, funcionários e professores.

Como em qualquer escola, existem ambientes que necessitam de acesso controlado, por exemplo, salas de professores, salas de aula, laboratórios onde ocorrem aulas, auditório e salas específicas de equipamentos ou computadores, totalizando cerca de 50 locais de acesso privado. Para acessar um desses locais citados, é necessária sua respectiva chave, sendo que existem no máximo duas

cópias de cada chave. Na portaria encontra-se um funcionário, que é responsável pelo armazenamento e registro de informações para controle dessas chaves mediante um caderno de registro que todas as pessoas que solicitam chaves, devem assinar. Este processo pode até ser simples, porém o CTISM possui uma equipe extensa de funcionários, que é representada por 67 professores, 29 funcionários técnico-administrativos e 44 bolsistas.

Pode acontecer uma situação na qual dois ou mais funcionários, por exemplo, Alice e Bob são professores de ensino médio e devem lecionar aula na mesma sala de aula. Alice deve ocupar a sala nos primeiros períodos da manhã. Então após o término de sua aula, Alice tranca a porta da sala e retorna a seu gabinete, esquecendo-se de devolver a chave na portaria.

Bob, saindo de seu gabinete, chega à portaria e solicita a chave para utilizar a mesma sala de aula, porém Alice não retornou a chave. Pior, além de sair da sala e esquecer a chave em seu bolso, teve de ir às pressas para sua casa em razão de alguma casualidade.

Neste contexto, surgem algumas questões. Como o Bob vai saber onde está a única cópia existente da chave daquela sala? Ou como saber se realmente Alice tem permissão para utilizar a sala de aula? Ou até mesmo como saber se a sala de aula não está ocupada, se a chave não se encontra na portaria? A resposta para as três perguntas é uma só: na implementação atual, realizada manualmente, não existe um modo confiável de garantir que esses eventos sejam solucionados de forma eficiente. Porém somente prestamos atenção nestas questões quando algo está errado.

Nesse contexto, é importante a existência de mecanismos que facilitem a implementação de autenticação, autorização e a auditoria. Autenticação para cada funcionário ter suas credenciais independentemente dos outros, removendo a limitação de uma pessoa por vez com a posse da chave de abertura da porta. Autorização, pelo fato de cada funcionário entrar apenas nas salas que seu nível de acesso permite, restringindo locais mais importantes a pessoas com maior nível de

acesso na instituição. E finalmente a auditoria, que permite verificar qual funcionário acessou um ambiente restrito e quando o fez.

1.2 Objetivo e metodologia

A finalidade dessa pesquisa é desenvolver um *software* para um sistema automatizado de controle de fluxo de pessoas, onde cada funcionário tenha suas credenciais, ampliando a mobilidade em questão das chaves, a segurança em virtude da autenticação e a auditoria no acesso a esses espaços reservados. O CTISM é usado como cenário de testes.

Mais precisamente, os objetivos específicos deste trabalho são:

- Projetar um *software* de controle de acesso composto de módulos que gerenciem as seguintes características:
  - Autenticação;
  - Autorização;
  - Auditoria;
- Baixo custo de implantação e manutenção;
- Implementar esse *software* de acordo com o projeto criado;
- Testar o *software* em ambientes controlados e em cenários reais de utilização do sistema (no CTISM).

O sistema ao qual se refere este projeto foi realizado para adicionar segurança e auditoria no controle a ambientes. Cada funcionário possuirá uma credencial própria, vinculada ao seu cadastro pessoal e com ela abrirá apenas as salas que estão no seu nível de acesso, estas possuindo dispositivos eletrônicos captadores de credenciais e controladores de acesso, como por exemplo, teclados numéricos e fechaduras eletrônicas. Com isso, o problema de funcionários sem acesso aos locais reservados será reduzido, adicionando ainda a auditoria

necessária, como a data e o horário que o funcionário entrou na sala, bem como todos que acessaram aquele mesmo local.

O projeto dos módulos que constituem o *software* deve ser independente do dispositivo de entrada ou da base de dados utilizada, entretanto, no escopo deste trabalho a interação entre os módulos é realizada em tempo de programação, embora em trabalhos posteriores exista a possibilidade da criação de *plug-ins* para adicionar os dispositivos e bases de dados diferentes.

A parte da implementação do *software* será realizada com recursos computacionais disponíveis nos laboratórios da Universidade Federal de Santa Maria e recursos físicos privados para a compra de dispositivos necessários para a implantação fora de um ambiente simulado de testes, de modo a validar o serviço.

1.3 Estrutura do texto

Este trabalho está dividido da seguinte maneira: o capítulo 2 descreve a revisão bibliográfica. O capitulo 3 expõe a visão geral, o *hardware* sobre o qual foi baseado o projeto a arquitetura do sistema e os métodos que o compõem de um modo mais específico. A seguir no capítulo 4, são comentados os testes com o *software* e seus resultados. Após essa parte, são apresentados os resultados obtidos por essa pesquisa e projetos futuros.

2 REVISÃO BIBLIOGRÁFICA

Ao passo que substituímos as chaves físicas dos ambientes por chaves digitais, a segurança da informação torna-se algo importante, pois com posse dela é possível usufruir de qualquer recurso dessa instituição. Torna-se necessário definir as características principais que constituem esse assunto.

Nesse capítulo serão estudados os princípios que definem o controle de acesso e seus mecanismos de controle e trabalhos relacionados, tanto proprietários quanto acadêmicos.

2.1 Controle de acesso: autenticação, autorização e auditoria

Segundo Santos (2007), a autenticação, a autorização e a auditoria em um sistema são essenciais para garantir que não ocorram acessos indevidos em aplicações e recursos de locais, que se mal gerenciados podem causar grandes problemas.

A autenticação é o ato de atestar como verdadeira a afirmação de que um sujeito é quem ele diz ser. Para que o usuário seja autenticado no sistema, é necessária uma combinação da identificação de usuário e de chave que seja única, pois caso contrário o sistema pode apresentar certo nível de ambiguidade no momento da verificação da identidade.

Santos (2007) ainda indica que a autenticação pode ser reforçada por meio de várias características como horários e localidades permitidas, por exemplo, computadores autorizados, catracas e portas.

Existem diversos métodos de um usuário autenticar-se em um sistema, por exemplo, nome de usuário e senha, biometria, *smartcards* e dispositivos autenticadores, também chamados de *One Time Passwords*, pois variam de tempos em tempos, como o utilizado na autenticação dos serviços do *Google* e da *Blizzard*

Entertainment, caso seja desejado. Esses métodos podem ainda ser combinados para um aumento de segurança.

A autorização geralmente é um processo que ocorre após a autenticação ser comprovada, onde o usuário autenticado recebe as permissões e restrições que lhe são devidas, de modo a utilizar as características do sistema de modo adequado.

No que se refere às políticas de atribuição de permissões e restrições de utilização dos recursos do sistema, ou seja, um conjunto de limitações e critérios, designados pelo administrador, de modo a limitar o acesso de um usuário apenas aos recursos necessários (IBM, 2011), existem vários métodos diferenciados para esse fim, porém serão citados os: MAC (*Mandatory Access Control)*, DAC (*Discritionary Acess Control)* e RBAC (*Role-Based Access Control)*.

Método MAC, é o método que consiste em atribuir níveis de acesso aos recursos e aos usuários, onde os usuários podem acessar os recursos menores ou iguais ao seu nível de acesso. É utilizado em locais que necessitam de forte segurança, por exemplo, laboratórios e instalações militares (SANTOS, 2008).

Método DAC: de acordo com Jordan (1987, p.3),

> O método DAC é um método de restringir o acesso baseado na identidade do sujeito ou grupos a que o mesmo pertence. Os controles são discricionários no sentido do recurso com certa permissão de acesso é capaz de passar essa permissão (talvez indiretamente) para qualquer outro recurso.

Diante disso, o método DAC utiliza o conceito de propriedade dos dados, pois a partir disso, o usuário pode restringir ou permitir o acesso a esses arquivos, por exemplo, permissões utilizadas com o comando *unix chmod*, que altera a permissão de leitura, escrita e execução de um arquivo ou diretório.

Método RBAC: segundo Barkley et al.(1997, p. 2):

> O controle de acesso baseado em papéis (RBAC) é uma alternativa às políticas tradicionais discricionárias (DAC) e obrigatórias (MAC) de controle de acesso. O RBAC foi criado para ser análogo, em questão de segurança, aos cargos dentro de uma empresa onde existem relações hierárquicas entre seus cargos e nunca um privilégio está vinculado diretamente a um usuário e sim aos cargos que ela exerce, com isso, bastando adicionar o usuário a esse grupo.

Este modo serve para remover a necessidade de adicionar permissões individuais aos usuários, ao invés disso, adicionar permissões aos grupos e inserir os usuários nestes grupos, herdando as permissões necessárias para a utilização dos recursos.

Devido à relevância deste método e as necessidades do CTISM, optou-se por utilizar o método RBAC, visto que uma escola separa as pessoas vinculadas em categorias, bastando apenas anexar as permissões e restrições a esses grupos.

2.2 Trabalhos Relacionados

Os trabalhos relacionados foram separados em duas categorias, soluções proprietárias e trabalhos acadêmicos.

2.2.1 Soluções proprietárias

As soluções proprietárias são produtos desenvolvidos por empresas de médio ou grande porte, geralmente especializadas na área de segurança. Essas empresas investem grandes quantidades de dinheiro em pesquisas e desenvolvimento de modo a propiciar sempre a melhor solução em segurança. A vantagem de utilizar uma solução dessas é o fato de ter uma compatibilidade maior entre equipamento provido pela empresa e o programa gerenciador de permissões de acesso. Porém existem várias desvantagens, como por exemplo, o produto ser vendido em um

“pacote” geralmente padrão, o código-fonte de o sistema gerenciador ser um código fechado ou o alto valor gasto para ter uma solução proprietária.

Dentre várias empresas, foram selecionadas algumas para verificação de alguns parâmetros, tais como auditoria, quantidade de dispositivos de diferentes modos de autenticação, situação do código-fonte e preço, se disponível.

Controle de Acesso Remoto Siemens. Segundo consta na página hospedada na *Internet* da Empresa Siemens (SIEMENS, 2011), esse controle de acesso permite cadastrar usuários, agendar horários para cada usuário e possui também características de auditoria, quando se refere ao fato de informar caso algum usuário entre em horário indevido. Também é possível vincular esse sistema a um sistema de alarme monitorado 24 horas.

Apesar de todas essas vantagens descritas, a página da empresa na Web apresenta apenas pouca informação relevante e três fotos, sendo duas demonstrando dispositivos de entrada das suas credenciais, teclado e cartão, respectivamente. A terceira foto apresenta um formulário, provavelmente interno ao *software* gerente. Não possui informação quanto a preço de contratação e cobra um valor extra na contratação de uma possível manutenção no sistema. O código-fonte dessa solução comercial não é citado em momento algum na página a que hospeda o produto.

Controle de acesso NibTec. A página dessa empresa na *Web* (NIBTEC, 2011) apresenta mais soluções em questão de dispositivos de entrada de credenciais. Existem opções desde teclados numéricos, passando por cartões com tecnologia de radiofrequência (RFID) até os dispositivos biométricos reconhecedores de impressões digitais, dispensando o uso de credenciais comuns.

Uma das diferenças dessa solução para a anterior é o fato de poder ser contratada para reforçar as medidas de seguranças tradicionais de um computador, complementando as credenciais comuns existentes em qualquer sistema operacional, tais como adicionar *scanners* de impressão digital em um computador.

Segundo a página da empresa, existe implementação de auditoria, pois a ferramenta gera relatórios de entrada e saída de usuários dos locais privados, apesar de não citar opções de agendamento de horários. Embora o *software* seja gratuito com a contratação do serviço, em momento algum consta a situação código-fonte. O preço de contratação do serviço também não é citado.

Controle de acesso ID TECH. O sistema de controle de acesso ID TECH (IDTECH, 2006), além de apresentar as soluções anteriores em dispositivos, também disponibiliza autenticação por impressões digitais ou reconhecimento de íris. A página da empresa também apresenta outras soluções para ambientes de grande porte, como penitenciárias, hospitais e prefeituras, de modo que provavelmente seja uma solução que possui auditoria.

Mais uma vez o código-fonte nem o preço são citados nas paginas referentes à solução.

Controle de acesso MADIS. Essa solução comercial (MADIS, 2011) também apresenta diferentes métodos de entrada de credenciais, como leitores de códigos de barras.

No quesito auditoria, a página indica que a partir do *software* gerente é possível agendar visitas, emitir relatórios individuais, e verificar o tempo de permanência de um usuário nos ambientes restritos. Não houve referências sobre código-fonte ou preço de contratação do serviço.

A análise dessas soluções comerciais, a grosso modo, indica que por serem empresas que fabricam produtos consumidos por clientes de grande porte, devem ocultar ao máximo seus códigos-fonte a fim de dificultar a busca por falhas no sistema e manter a propriedade intelectual devido ao alto custo de contratar tal serviço.

2.2.2 Soluções acadêmicas

Sentinel: um engenho Java para controle de acesso RBAC. Esta solução acadêmica, implementada no ano de 2003, pelo até então aluno Cristiano Mattos da Universidade Federal de Pernambuco, descreve um sistema em Java para controle de acesso baseado em papéis (Mattos, 2003).

Por exemplo, em uma empresa existem vários grupos organizacionais, com relações hierárquicas entre eles. Dentre estes grupos, são selecionadas duas categorias: infraestrutura de rede e o grupo dos telefonistas. O grupo da infraestrutura de rede possui privilégios distintos do outro grupo, pois necessitam entrar em locais onde passam os equipamentos de redes para manutenção. O grupo dos telefonistas possui a função de atender seu telefone, realizar anotações para os outros usuários e outras atividades semelhantes.

Visto que os grupos não compartilham privilégios, talvez seja necessário que um funcionário do grupo da infraestrutura atenda as chamadas e realize anotações sobre os problemas existentes de rede. Ao invés de estender o privilégio necessário a este funcionário da empresa, o funcionário é inserido no grupo dos telefonistas, retendo por herança ambos os privilégios dos grupos.

Nessa solução acadêmica, o autor cita a existência de um módulo de auditoria que registra os acontecimentos, porém explicita a simplicidade desse módulo, pois o trabalho é focado na autenticação. O autor não vincula um dispositivo de entrada de credenciais em particular ao projeto, talvez pelos testes não utilizarem em um cenário real (Mattos, 2003).

Nas suas considerações finais do trabalho, esse autor ainda salienta o fato de ser utilizado o CORBA nos projetos futuros em virtude da robustez desse *middleware*.

Desenvolvimento de uma interface web para integração e configuração da rede de sensores-atuadores do projeto CONVERGE UFSM. Este trabalho de graduação realizado pelo aluno Cristiano Cortez da Rocha, para a conclusão do

curso de Ciência da Computação em 2007, propõe a criação de uma interface web para a administração dos recursos gerenciados pelo projeto CONVERGE. Entretanto, aqui será abordado o método de autenticação deste sistema e o gerenciamento utilizado pelo autor em seu trabalho para integrar seus recursos de interesse.

ROCHA (2007), utilizando a política de controle de acesso baseado em papéis (RBAC), demonstra em seu trabalho a relevância de coletar informações acerca dos tipos de usuários e recursos utilizáveis, de modo a inserir esses usuários em grupos distintos, inicialmente chamados de usuários comuns e usuários administradores, onde o nível desses usuários indica seu acesso aos recursos segundo a tabela 1.

Tabela 1 – Tipos de usuários e suas permissões no sistema

Permissões	Usuário Administrador	Usuário Comum
Abertura de porta	Sim	Sim
Edição de configuração de eventos/ações	Sim	Não
Visualização de eventos	Sim	Não
Visualização de logs	Sim	Não
Visualização *on-line* de vídeo	Sim	Sim
Troca de senha de acesso	Sim	Sim

Fonte: (Rocha, 2007)

Como visto na tabela, a política RBAC proporciona uma facilidade no quesito administração das permissões e restrições, pois os usuários herdam as permissões dos grupos aos quais eles pertencem.

Na área de auditoria, ROCHA (2007) ainda afirma existir um sistema para registro de ponto, possibilitando a criação de relatórios. Existe a possibilidade da verificação de *logs* do sistema, pois a utilização dos recursos do sistema dispara o registro dessa ação.

2.3 Conclusão parcial

Com base nesses sistemas estudados, verificou-se que é interessante desenvolver um sistema para o controle de acesso com uma implementação própria, embora muitas características destes sistemas possam ser aproveitadas, adicionando outras funcionalidades necessárias ao contexto escolar cujo sistema será instalado. O sistema a ser desenvolvido deverá atender às necessidades do CTISM. A tabela 2 faz comparações entre os sistemas, proprietários e acadêmicos.

Tabela 2 – Comparação entre os diversos sistemas considerando características específicas.

	Siemens	NibTec	ID Tech	MADIS	Sentinel	Converge
Auditoria	Registra horário de acesso, monitoria	Relatórios de entrada e saida	Provavelmente possui auditoria bem implementada	Relatórios individuais, tempo de permanência e agendar visitas	Registra horário do acesso	Registro de ponto, relatórios e logs do sistema
Dispositivos físicos	Teclado numérico	Teclado numérico, RFID, impressão digital	Teclado numérico, RFID, impressão digital, íris	Teclado numérico, RFID, código de barras	Não consta um método padrão	Não consta
Custo	Não consta	Não consta	Não consta	Não consta	Livre	Livre
Disponibilidade de código-fonte	Não	Não	Não	Não	Trechos de código	Não

3 SISTEMA PARA CONTROLE DE ACESSO AO CTISM

Neste capitulo apresentamos mais detalhadamente o sistema de controle de acesso ao CTISM. Este capítulo está dividido em quatro seções: visão geral do sistema, *hardware* utilizado no projeto, a arquitetura do sistema e uma explicação mais aprofundada dos módulos que o compõe.

3.1 Visão geral do sistema

O sistema basicamente pode ser visto como um algoritmo produtor-consumidor, onde os dispositivos conectados às portas recebem credenciais, anexam esses eventos em filas definidas e despejam o conteúdo das filas em uma fila principal, onde um consumidor aguarda a chegada dos eventos para resolvê-los. De acordo com a categoria do evento, definido no momento de sua criação, o consumidor é capaz de analisá-lo e tratá-lo de modo adequado.

Após a verificação da categoria desse evento, o consumidor gera uma conexão com a base de dados, verificando as informações contidas nesse evento e retornando ao dispositivo responsável a resposta dessa verificação. São criados registros da tentativa de acesso, independentemente do resultado.

É importante ressaltar que o sistema se comunica com a base de dados e com o dispositivo de entrada de credenciais pelo protocolo TCP/IP, podendo ser utilizado o cabeamento de Internet já existente no local. Essa característica torna a utilização mais fácil, visto que a maioria dos teclados é conectada via portas seriais.

3.2 Hardware utilizado

O *hardware* que existia anteriormente ao sistema a qual esse trabalho se refere foi um teclado desenvolvido artesanalmente no laboratório do Serviço de Suporte à Informática (SSI) do CTISM, para fins de controle de acesso às portas. A esse projeto, foi adicionado um componente chamado PIC *Mini Web* apresentado na figura 1.

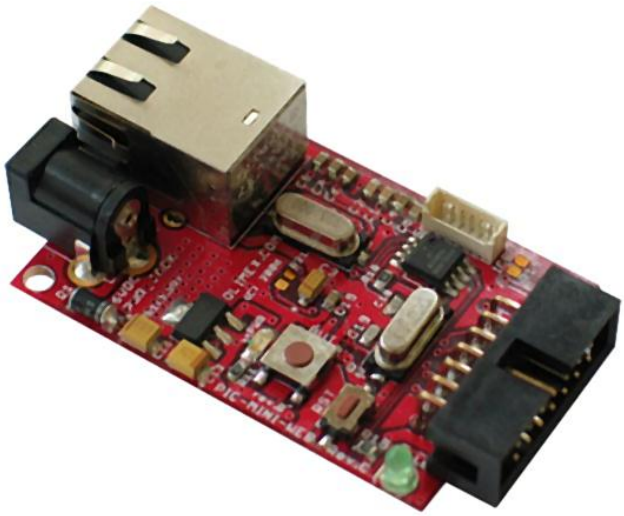

Figura 1 – Placa com microcontrolador PIC e sua entrada RJ-45
Fonte: http://www.olimex.com/dev/pic-mini-web.html

O PIC *Mini Web* possui uma porta RJ-45 de 10 *megabits* de conexão, 32KB de memória Flash para o programa, 1MB de memória de armazenamento para páginas web e 1KB de memória RAM. O software já incluso de fábrica possui a pilha de protocolo TCP/IP implementada e adaptada ao PIC, além de servidores HTTP, FTP e TELNET. Esta placa é conectada ao dispositivo controlador, neste caso a fechadura elétrica da porta.

O servidor que hospeda o *software* possui 2GB de memória RAM, 250 GB de disco rígido e executa o sistema operacional Ubuntu 10.10.

3.3 Arquitetura do sistema

A arquitetura do sistema é dividida em cinco módulos. Os módulos consistem em:

- Módulo de conexão com o dispositivo de entrada (1);
- Módulo gerente (2);

- Módulo de auditoria (3);
- Módulo de conexão com a base de dados (4);
- Módulo de administração do sistema (5).

Para auxiliar essa tarefa, serão apresentados diagramas para representar graficamente as relações entre os módulos. A figura 2 representa os módulos e suas conexões.

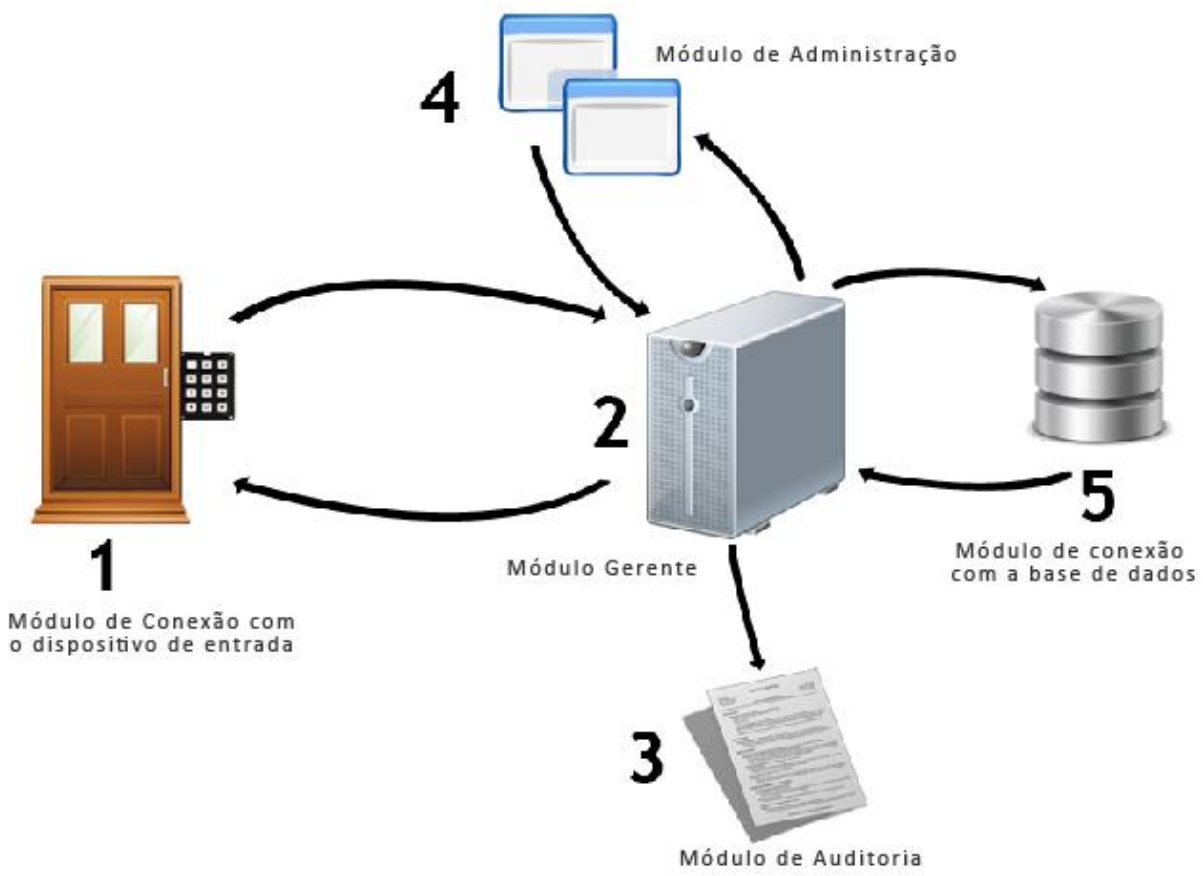

Figura 2 - Módulos do sistema

3.3.1 Módulo de conexão com o dispositivo de entrada

Conforme mencionado anteriormente, o processo de conexão com o dispositivo começa quando um funcionário manifesta a intenção de entrar em um local protegido pelo dispositivo controlado pelo sistema. Esse funcionário fornece suas credenciais e a partir dessas, uma área do dispositivo de entrada codifica as informações do usuário e as prepara para envio.

A seguir, o dispositivo informa ao módulo gerente que ele está pronto para o envio dos dados, conforme a figura 3:

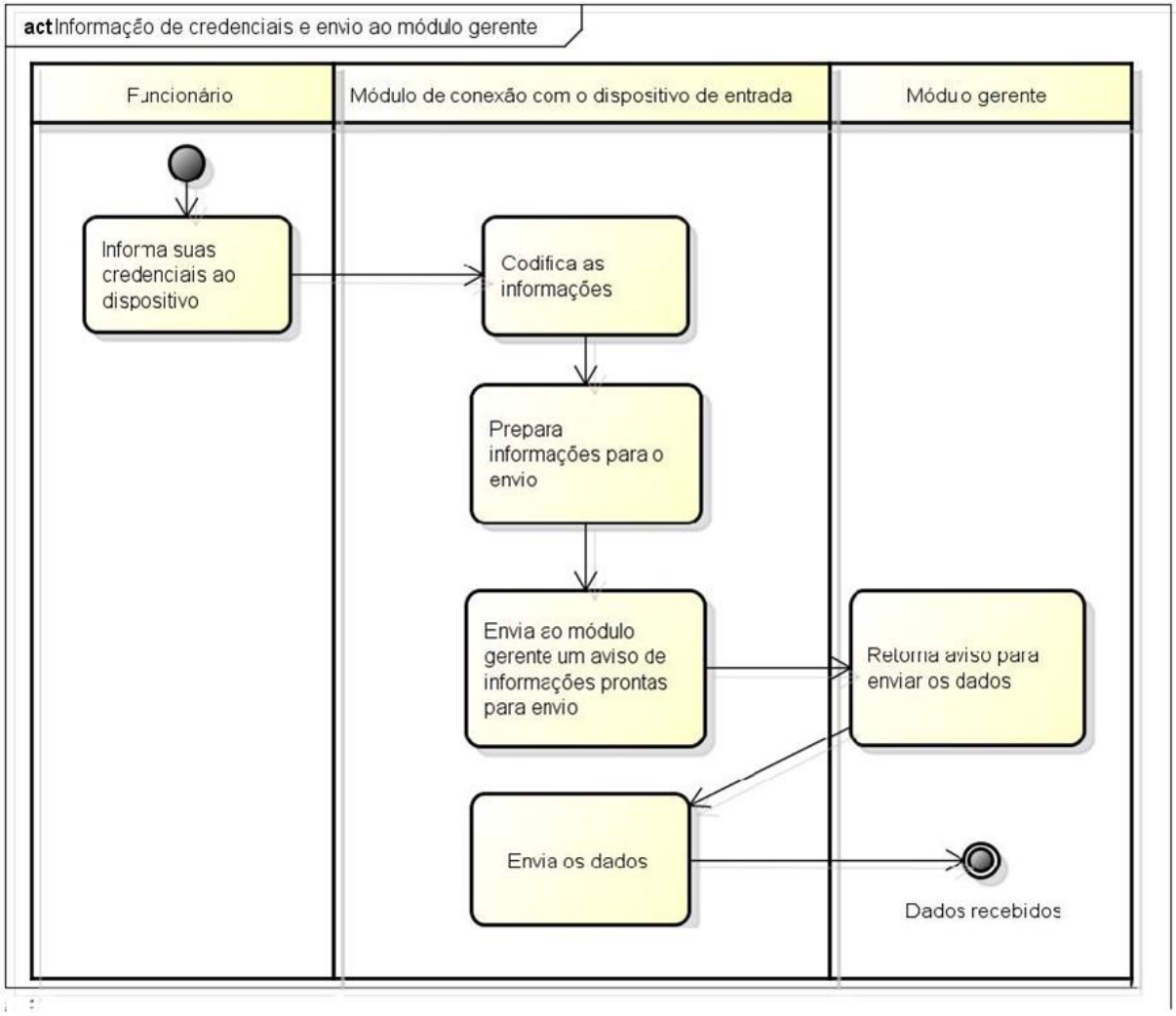

Figura 3 - Informação de credenciais e envio ao módulo gerente

O módulo de conexão com o dispositivo também recebe a informação do módulo gerente para abrir a porta ou mantê-la fechada.

3.3.2 Módulo gerente

Esse módulo gerencia as ações de todos os outros módulos. A partir das configurações realizadas no módulo de administração do sistema, o módulo gerente é carregado com os endereços dos dispositivos físicos de abertura de porta e da base de dados, para transferência de informações.

No momento que o módulo gerente recebe as informações do módulo de conexão com o dispositivo de entrada, estas são decodificadas e organizadas para envio ao módulo de conexão com a base de dados. Com base nas configurações estabelecidas no módulo de administração do sistema, essas credenciais são enviadas à base de dados para verificação de permissão daquela porta.

O módulo gerente recebe o resultado da verificação, registra a resposta de modo apropriado utilizando métodos específicos do módulo de auditoria e a encaminha ao módulo de conexão com o dispositivo, de acordo com a figura 4.

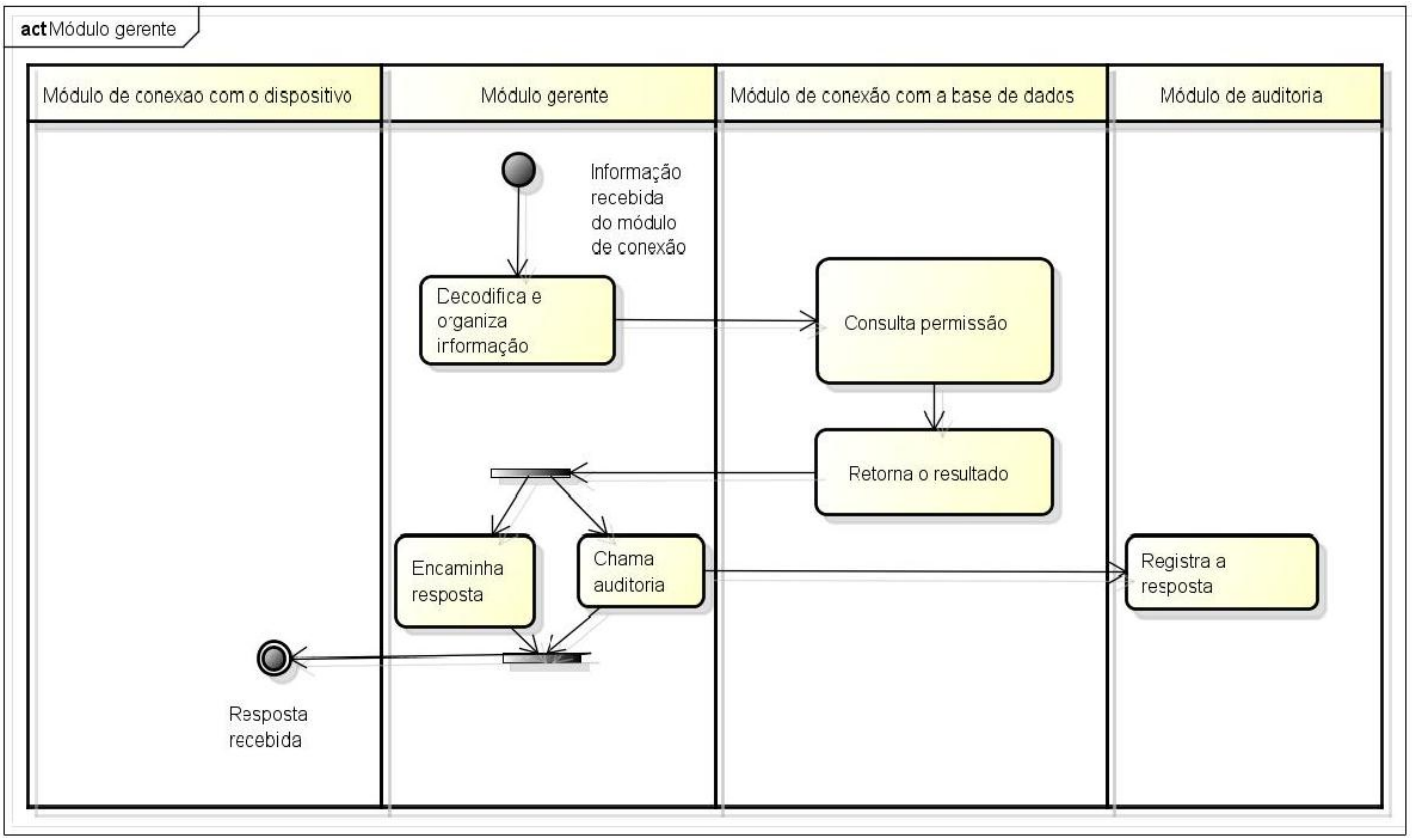

Figura 4 - Atividades do Módulo gerente

3.3.3 Módulo de auditoria

Este módulo é o responsável por gravar todas as ações que os funcionários executam, tanto ao inserir suas credenciais no dispositivo de entrada, quanto ao acessar o módulo de administração do sistema. Existem dois tipos de arquivos de registros: de acesso às portas e de acesso e alteração do módulo de administração do sistema.

O registro de acesso às portas armazena informações referentes às intenções de acionamento do dispositivo, como por exemplo, a ação de um funcionário ao informar suas credenciais ao dispositivo que controla a porta. Nesse caso, o módulo de auditoria armazena as credenciais desse funcionário, a data e a hora que a tentativa foi realizada, o número de identificação da porta e o resultado (entrada bem sucedida ou falha de autenticação).

Com essas informações de registro é possível obter um mapeamento gráfico, demonstrando o caminho que o funcionário percorreu em um intervalo definido de tempo (i.e. turno, dia, etc.) e também se pode obter a localização aproximada de um funcionário, desde que ele esteja usando uma sala protegida, pois o sistema atualiza as ações que cada funcionário executa no sistema.

No registro de acesso e alteração do módulo de administração, são armazenados dados referentes ao acesso ao módulo de administração, contendo data e hora, informação de credenciais, assim como as alterações realizadas pelo funcionário nas configurações de administração do sistema.

3.3.4 Módulo de conexão com base de dados

Este módulo é responsável por receber as informações vindas do módulo gerente, estabelecer a comunicação com a base de dados e a consulta, retornando um evento com o resultado ao módulo gerente, de acordo com o dispositivo utilizado pelo funcionário.

No momento em que o módulo de conexão à base de dados recebe a informação, é estabelecida uma conexão com essa base. Quando essa conexão é estabelecida, o módulo de conexão à base de dados extrai as informações necessárias para verificação e realiza a consulta. Esse módulo retorna um evento contendo a resposta ao módulo gerente, em seguida finaliza a conexão com a base de dados.

Como existem várias plataformas distintas de bases de dados, por exemplo, LDAP ou SQL, as configurações que definem a base a ser utilizada e o modo da conexão de dados se encontram no módulo de administração do sistema.

3.3.5 Módulo de administração

Neste módulo encontram-se todos os parâmetros necessários ao funcionamento do sistema, subdivididos em áreas. Existem seis áreas disponíveis aos funcionários que possuem acesso: área de configuração de dispositivos de

entrada, área de configuração de base de dados, área referente a permissões e restrições de funcionários, área de auditoria do sistema, área de agendamento de horários e área de ajuda ao usuário do sistema.

3.4 Aspectos internos dos módulos do sistema

Esta seção aborda tópicos estruturais da codificação dos módulos que constituem o *software* controlador de acesso.

Os módulos foram codificados em Java, linguagem selecionada por executar seus programas em uma máquina virtual JVM, possibilitando independência do sistema operacional utilizado e também pela capacidade de mesclá-la com linguagens *web* para uma utilização através da internet sem maiores dificuldades. Ao longo desta seção, serão inseridas imagens para ilustrar as classes utilizadas que compõem esses módulos.

Entretanto, antes de entrar diretamente nos detalhes dos módulos, é necessário abordar a classe *Event,* utilizada no sistema para transferência de informações entre os módulos. No instante que um funcionário informa suas credenciais a algum dispositivo de entrada, objeto dessa classe é criado e repassado para uma fila específica. Esta classe possui os seguintes atributos:

- tipo, que é um variável do tipo *String,* para representar o tipo de conector utilizado;
- objeto, que é uma variável de qualquer tipo, destinada a armazenar as informações das credenciais do funcionários;
- conector, um ponteiro que referencia o dispositivo de entrada utilizado.

3.4.1 Módulo de conexão com o dispositivo de entrada

Este módulo utiliza como referência uma estrutura do tipo interface, ou seja, “uma interface define um novo tipo de referência. Entretanto, diferentemente de outras classes, interfaces não provêm implementações para os tipos que definem.”

(FLANAGAN, 2005, p. 135). Pelo fato da interface ser uma classe de métodos abstratos, para utilizar um dispositivo de entrada, é necessária a criação de uma classe que herde os métodos que fazem a comunicação de um módulo a outro, conforme a figura 4.

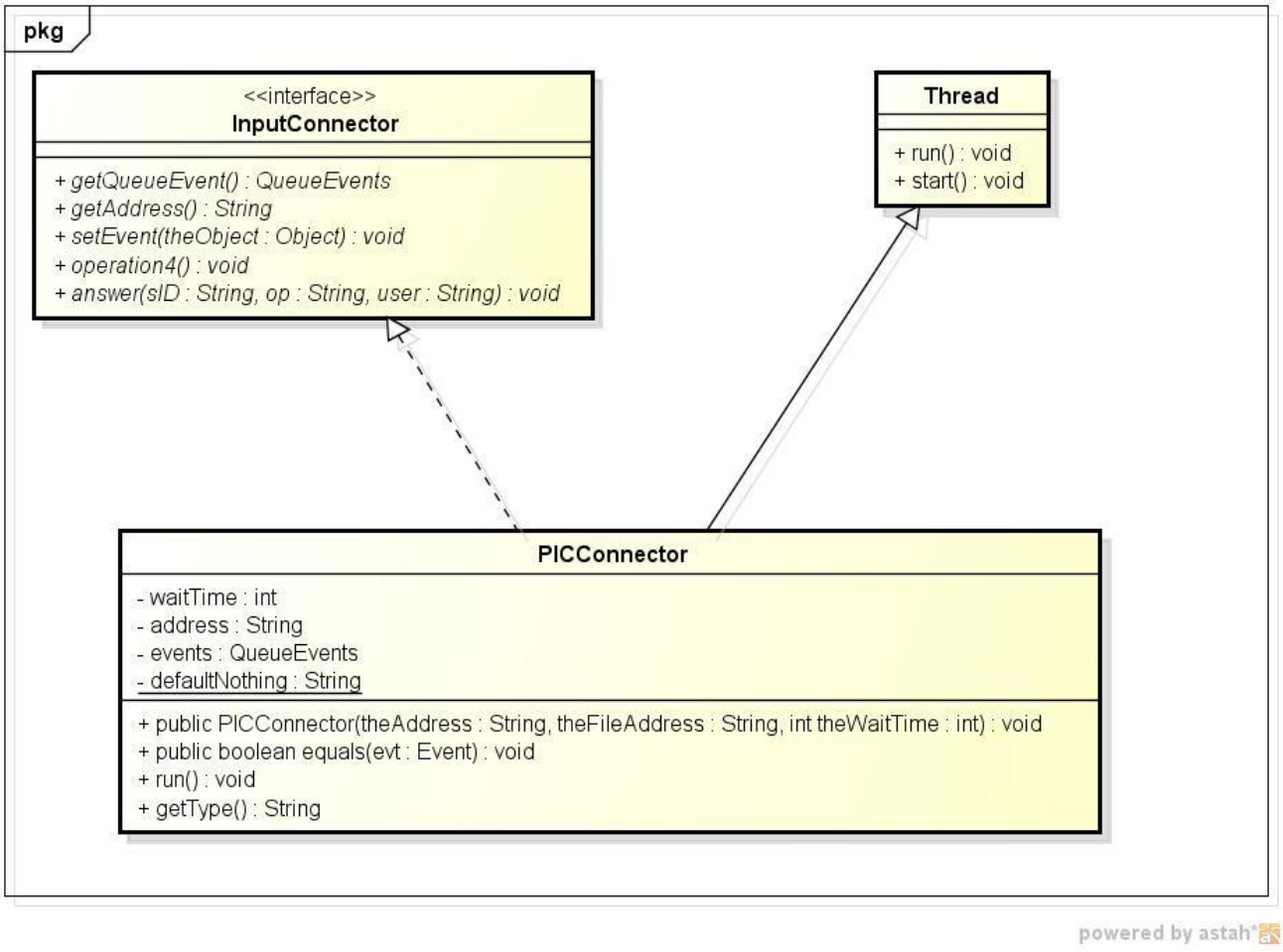

Figura 5 - Diagrama de classes do módulo de conexão com o dispositivo de entrada.

A figura 5 mostra a criação de uma classe para a conexão de um dispositivo de entrada chamada de *PICConnector.* Esta classe implementa os métodos herdados por duas classes, a interface *InputConnector* e a classe *Thread.*

Para a transferência dos dados de forma adequada ao módulo gerente, os métodos herdados da classe *InputConnector* devem ser complementados por métodos criados na própria função *PICConnector*.

A classe *Thread* é utilizada pela sua versatilidade em executar várias instruções concorrentemente, de modo que o programa não aguarde a resolução de uma instrução que para ele seria irrelevante em dado momento. O método *run* é sobrescrito para, neste tipo de conector, utilizar uma estratégia baseada em *Polling,* isto é, verificando a atualização de uma informação a cada tempo determinado pelo usuário que também pode ser chamada atualmente também como *busy waiting,* devido à utilização de múltiplos threads competindo pelo processador (Hailperin, 2007).

Esta abordagem foi utilizada porque o microcontrolador PIC-Mini-Web para o qual este código foi criado possui um servidor *web,* tornando necessária a verificação da atualização da hospedada neste servidor.

Um exemplo dessa estratégia baseada em *polling* encontra-se na figura 6:

```
funcao run {
   URL endereco = Conector.endereco
     enquanto condicão_verdadeira
        conexao = gera_conexao_html(endereco)
        conteudo_da_pagina = le_informacao_da_pagina(conexao)
        se conteudo_da_pagina != conteudo_padrao
           evento = gera_evento(conteúdo_da_pagina)
           se evento.igual(evento_no_fim_da_fila)
            adiciona_na_fila(evento)
           fim se
        fim se
        conexao.fecha_conexao_html()
     fim enquanto
}
```

Figura 6 – Função *run* que implementa o *polling*

O método *answer* encontrado na interface *Connector,* é implementado de modo a retornar a resposta da consulta às informações para o servidor *web*,

ativando os mecanismos atuadores necessários e permitindo ou não a passagem do funcionário, segundo a figura 7.

```
funcao answer(String sID,String op, String user) {
     URL endereco =
          Conector.endereco + "/" + sID + "/" + op + "/" + user
     conexao = gera_conexao_html(endereco)
     conexao.fecha_conexao_html()
}
```

Figura 7 – Função *answer,* que envia a resposta ao dispositivo de entrada.

A variável sID é o número de identificação de sessão entre o dispositivo de entrada e o módulo gerente, afim de impedir comunicações externas, *op* é a operação utilizada e *user* é o nome do usuário que inseriu suas credenciais.

Como o *PICConnector* apenas hospeda a informação para ser recolhida pelo *polling,* existe uma preocupação acerca da segurança desses dados. A ideia inicial é utilizar uma encriptação robusta, porém ao longo do projeto é possível notar as limitações dos sistemas embarcados. Foi então necessário desenvolver um algoritmo que dificultasse a interpretação dos dados em caso de visualização indevida, utilizando a quantidade de recursos que esse sistema disponibilizava. Essas limitações serão explicadas no capítulo de *Hardware* desse trabalho.

Neste contexto, a classe *Crypt* recebe os dados da credencial do funcionário e embaralha com o Id da sessão, operações XOR de modo a dificultar a visualização das informações do usuário. Uma implementação da classe *Crypt para credenciais de* três caracteres e id da sessão de dois caracteres encontra-se na figura 8.

```
funcao Crypt(String info, int id_de_sessao, int numero){
   String resultado
   resultado[0] = info[2]
   resultado[1] = sID[0]
   resultado[2] = info[0]
   resultado[3] = info[1]
   resultado[4] = sID[1]
   int temporario = torna_inteiro(resultado)
   temporario = XOR(temporario, numero)
   retorna torna_string(temporario)
}
```

Figura 8 – Exemplo de função de codificação de dados

No caso de um usuário que colocasse como credenciais os valores 1,2 e 3, em vez de mostrar esses valores em texto plano, o dispositivo enviaria para seu servidor *web* a seguinte informação:

38598

onde o id da sessão é igual a 85 e a variável chamada “numero” igual a 555, tendo como resultado final um aumento na segurança da informação.

3.4.2 Módulo Gerente

Este módulo é composto de diversas classes, cada classe desempenhando um papel essencial no sistema, porém apenas uma instância de cada era necessária. Foi utilizado o *Design Pattern* do tipo *Singleton,* pois de acordo com Metzker e Wake (2006, p. 81), “a intenção do S*ingleton* é assegurar que uma classe tenha apenas uma instância e promover um ponto de acesso global a ela”, servindo perfeitamente aos propósitos do sistema. Cada classe especificada executa suas instruções concorrentemente, estendendo a classe *Thread* e sobrescrevendo o método *run.* A figura 9 ilustra o funcionamento do módulo gerente:

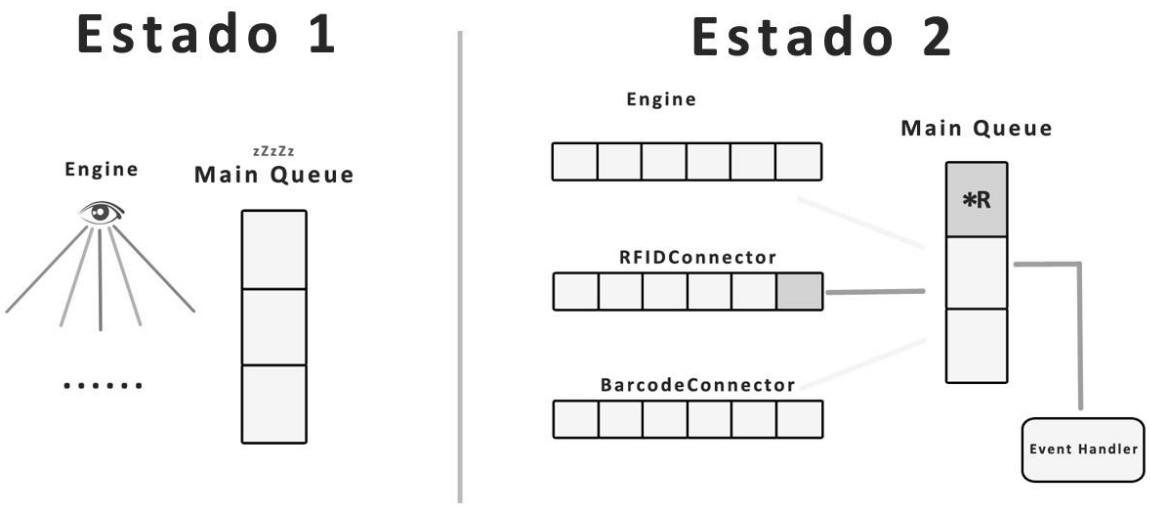

Figura 9 – Interação das classes do módulo gerente

No “Estado 1” da imagem anterior (Figura 9), *Engine* aguarda a chegada de novos dispositivos de entrada, enquanto *MainQueue* se mantém em estado de espera. No “Estado 2”, *Engine* já notificou *MainQueue* da existência de dispositivos de entrada, começando assim a busca por novos *Events* a serem enviados para o *EventHandler.*

- Classe *MainQueue*

Esta classe é a fila principal do sistema, onde internamente existe uma lista de ponteiros para filas de eventos enviados pelos dispositivos de entrada. Basicamente ela busca circularmente eventos nos dispositivos de entrada para posterior avaliação. Ela é também responsável pela adição e remoção sincronizada das filas desses dispositivos, pois são utilizados concorrentemente gerando a necessitando uma zona de exclusão mútua para manipulação desses dados.

- Classe *EventHandler*

EventHandler é a classe responsável por analisar os eventos e encaminhá-los à consulta de acordo com a base de dados selecionada e com o tipo de dispositivo de entrada. No momento que o evento é enviado para análise, a classe verifica o tipo de objeto que criou este evento, pois existem diferenças nos dados enviados

pelos dispositivos de entrada. Por exemplo, quando utilizado um teclado que necessita de *login* e senha, são enviadas essas duas informações e também o id da sessão. Quando utilizado um *smartcard,* são enviados apenas a *Tag* do cartão e o id da sessão.

Esta classe possui uma função chamada avaliaTipo, que reconhece o tipo de dispositivo que gerou o evento, executa o método adequado de consulta e envia ao dispositivo de entrada a resposta dessa consulta, permitindo ou não a passagem do funcionário pela porta protegida. A figura 10 demonstra a implementação do método *run* da classe *EventHandler*.

```
funcao run(){
   int i = 0
   enquanto(condicao_verdadeira){
      enquanto i<MainQueue.size(){
         QueueEvents temporaria = MainQueue.get(i)
         Event evt = temporária.pop()
         avaliaTipo(evt)
      }
   }
}
```

Figura 10 – Pseudocódigo da função principal da classe *EventHandler.*

- Classe *Engine*

Esta classe é a responsável por avisar às *threads* que já existem conectores cadastrados. Essa operação é necessária devido à natureza concorrente de todo o sistema. Por operar em um contexto sincronizado, o MainQueue necessita que o *Engine* verifique a disponibilidade de conectores. Em caso positivo, avisa à classe *MainQueue* que ela já pode voltar a procurar por eventos criados. Em caso negativo, apenas mantém a *MainQueue* em espera.

3.4.3 Módulo de auditoria

Este módulo é utilizado para registrar em memória persistente os movimentos dos usuários no sistema, tanto nos dispositivos de entrada, quanto no acesso ao módulo de administração.

- Registro de ações no sistema

No que se refere à utilização dos dispositivos de entrada, no momento da resposta à consulta, o sistema solicita a esse módulo um registro do nome do usuário, da localização e da resposta, positiva ou negativa. No caso da utilização do módulo de administração, o sistema após realizar as modificações desejadas pelo funcionário, avisa ao módulo de auditoria as alterações feitas. Em ambos os casos, as informações são registradas em uma base de dados, em tabelas diferentes, para facilitar uma consulta posterior.

- Controle de acesso baseado em papéis

O módulo de auditoria implementado utiliza a política RBAC, pois suas características adicionam o fator de segurança desejado ao sistema. O sistema promove a criação de grupos com permissões diferentes de utilização dos recursos conforme a tabela 3.

Tabela 3 – Grupos de usuários no sistema

Permissões	Administrador	Professor	Funcionário	Aluno
Sala SSI	Sim	Não	Não	Não
Cozinha	Sim	Sim	Sim	Não
Sala dos servidores	Sim	Não	Não	Não
Laboratórios	Sim	Sim	Sim	Não
Relógio-ponto	Sim	Sim	Sim	Não

Com a criação dos grupos, pode ser eliminada a necessidade de atribuir o acesso a cada recurso ao funcionário, apenas necessitando adicioná-lo ao grupo.

- Geração de relatórios

Os relatórios podem ser gerados em tela ou impressos de acordo com filtros específicos tais como “porta utilizada”, “usuário” ou “data e intervalo de horários”, retornando as informações relevantes.

3.4.4 Módulo de conexão com a base de dados

Assim como o módulo de conexão com o dispositivo de entrada, este módulo é utilizado através de uma interface, portanto para cada base de dados diferente é necessária uma classe que implemente os métodos dessa interface. Neste sistema atualmente foram implantados dois tipos de bases de dados, MySQL e LDAP. O primeiro é um sistema de gerenciamento de banco de dados do tipo relacional enquanto o segundo é um serviço de diretório. O que difere nos dois casos são as instruções da consulta, pois o retorno é o mesmo em ambos.

Na figura 11, temos o diagrama de classes de ambas classes implementadas utilizando a interface *DBConnector*:

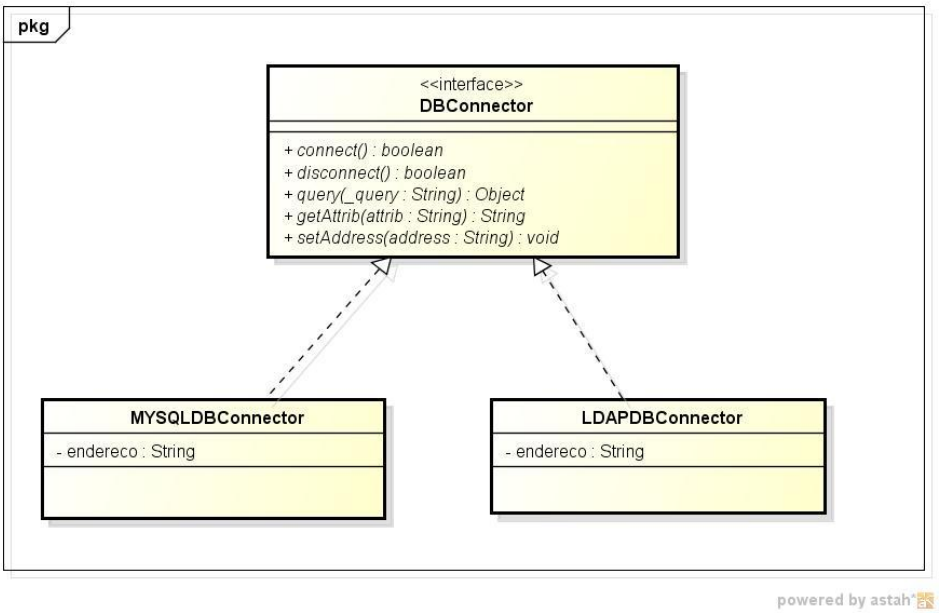

Figura 11 – Interface *DBConnector* e as classes que a implementam.

3.3.5 Módulo de administração

Este módulo coordena a configuração de todo o sistema. Feito em ambiente gráfico, o módulo de administração possui seis botões representando as áreas a serem administradas. Para ingressar na área de administração, é necessário que o usuário tenha suas credenciais cadastradas e deve pertencer a pelo menos um grupo que possua essa permissão de acesso. A partir disso o usuário deve utilizar suas credenciais na tela de *login* mostrada na figura 12.

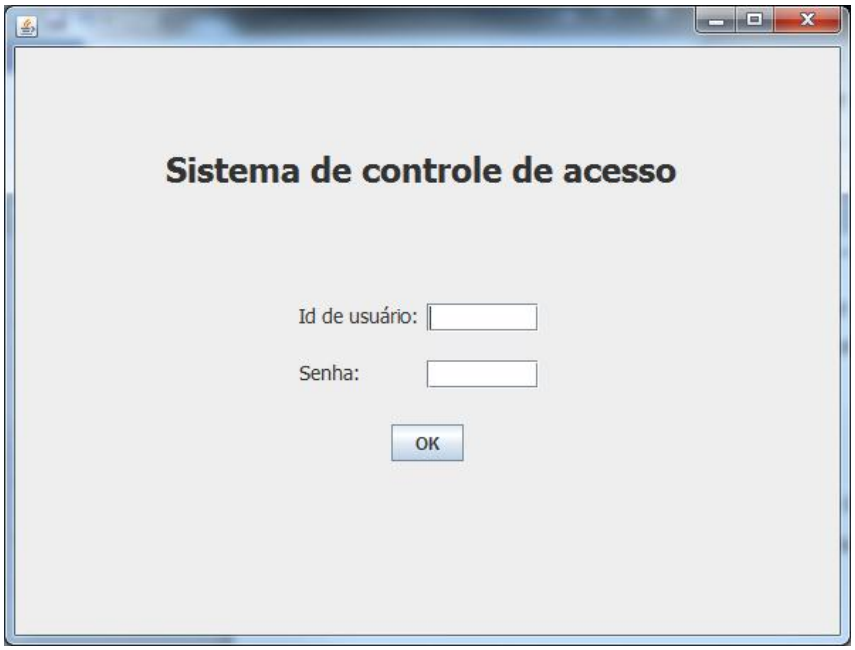

Figura 12 – Tela de *Login* do sistema

A partir da tela de *login* o usuário acessa um menu com as opções de administração do sistema.

- Conectores

A área dos conectores é o local onde são adicionados ou removidos dispositivos de entrada. Cada dispositivo de entrada pode ter seu endereço anexado para facilitar uma posterior verificação, conforme a figura 13. Quando adicionado, o dispositivo passa a funcionar automaticamente, a menos que seja parado manualmente, removido ou esteja fora de seu horário de funcionamento. No escopo deste trabalho, os conectores são implementados em tempo de compilação, porém em trabalhos futuros é possível a implementação de *plug-ins* para conectores.

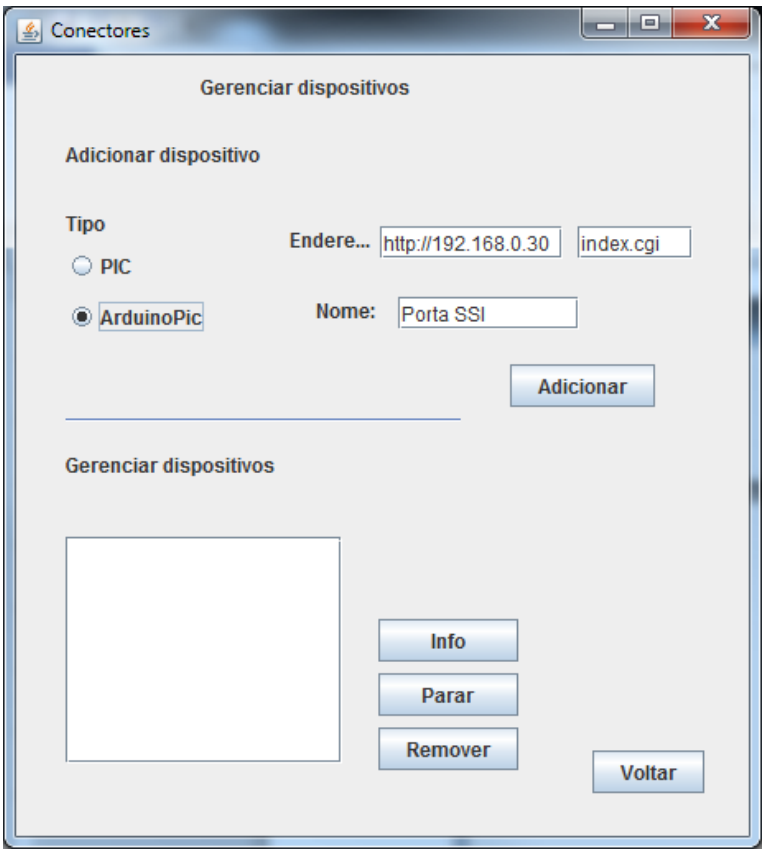

Figura 13 – Tela de gerenciamento de dispositivos de entrada

- Usuários

Na área de administração de usuários, é possível adicionar, remover e editar os usuários pertencentes ao sistema, incluindo adicioná-los e removê-los dos grupos pré-cadastrados no sistema, de acordo com a figura 14. Em todas as atividades utilizadas na área de gerenciamento de usuários, o módulo de conexão com a base de dados é chamado para atualizações dos usuários e para registro de logs para auditoria.

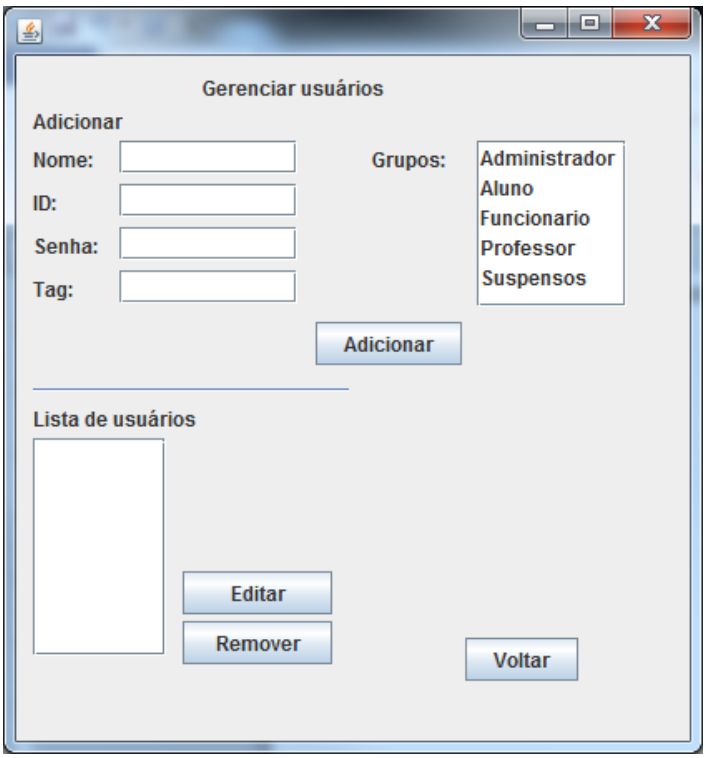

Figura 14 – Tela de Gerenciamento de usuários

- Auditoria

Neste local, são realizados relatórios e consultas baseados nos registros do sistema. É possível estabelecer filtros baseados em pessoas, horários e locais. Novamente é utilizado o módulo de conexão com a base de dados, a fim de receber os dados solicitados. A interface gráfica da área de auditoria é apresentada na figura 15.

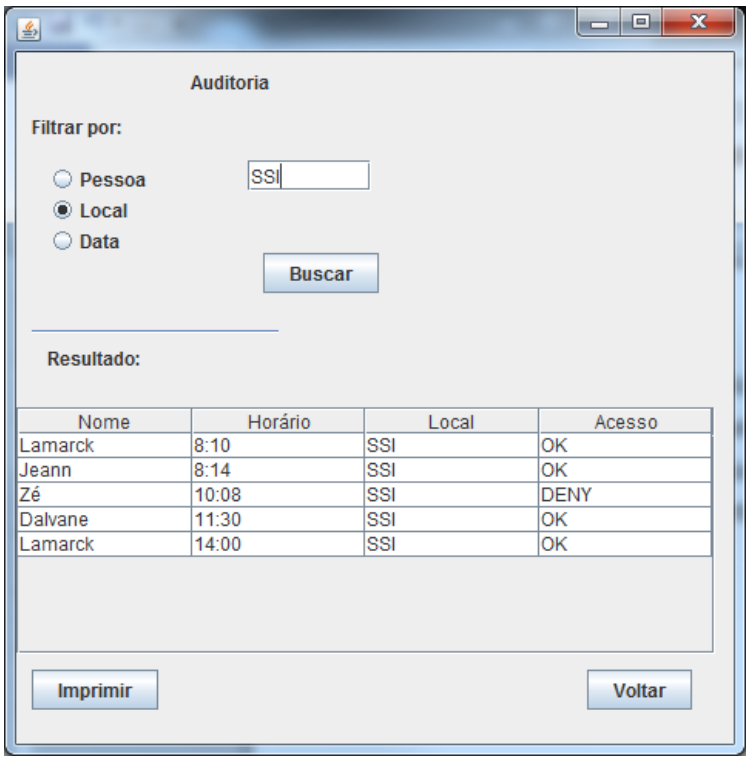

Figura 15 – Tela de gerenciamento de auditoria

- Base de dados

A área de base de dados é responsável pela configuração do local onde serão armazenados os registros e as informações de usuários. Aqui é possível selecionar uma entre os diferentes tipos de bases de dados disponíveis. Entretanto, a implementação atual contempla apenas as opções LDAP e MySQL, existindo também a possibilidade de utilizar o sistema que execute seus serviços sobre SSL. A tela com estas opções encontra-se na figura 16.

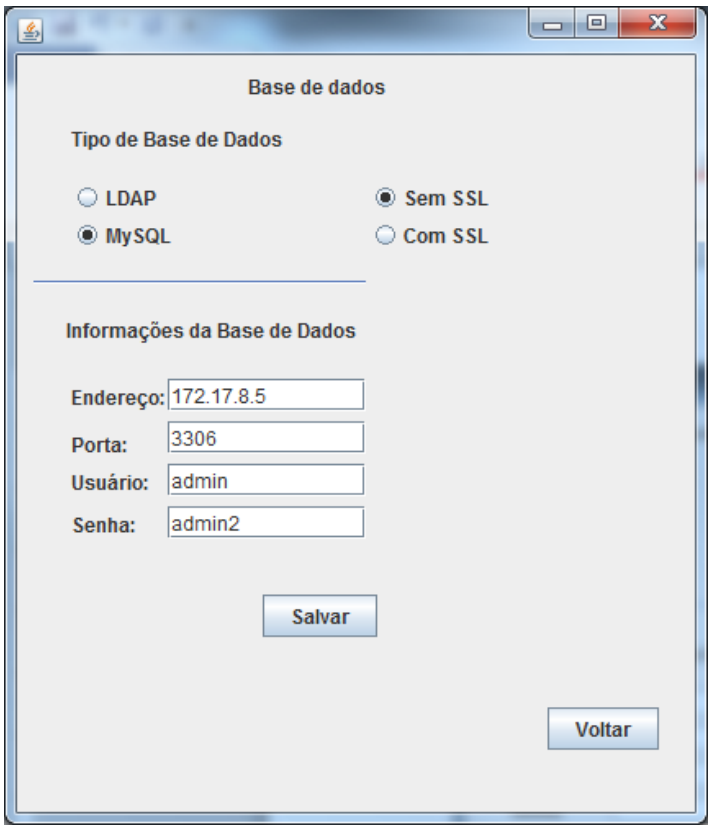

Figura 16 – Tela de gerenciamento de base de dados

- Grupos

Na tela de gerenciamento de grupos é permitido criar, remover e editar os grupos conforme a necessidade do sistema. É possível visualizar os usuários pertencentes aos grupos e as portas que esses grupos têm acesso. A interface da tela de gerenciamento de grupos está disponível na figura 17.

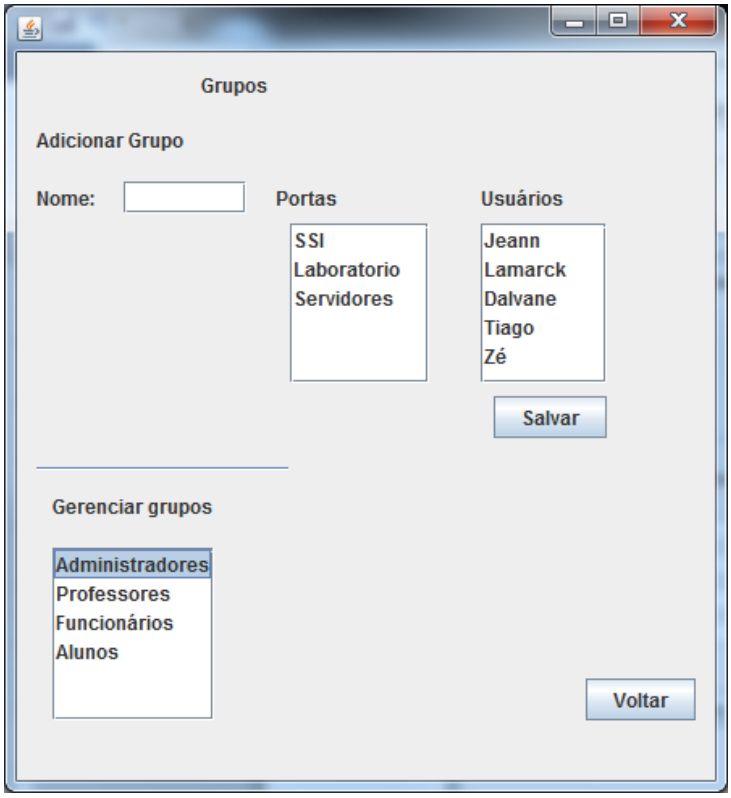

Figura 17 – Tela de gerenciamento de grupos

- Ajuda

O local de ajuda é um ambiente onde os usuários procuram por instruções para sanar suas dúvidas. Aqui existem informações para os procedimentos referentes ao sistema. Por exemplo, um usuário deseja cadastrar um grupo, conforme a figura 18.

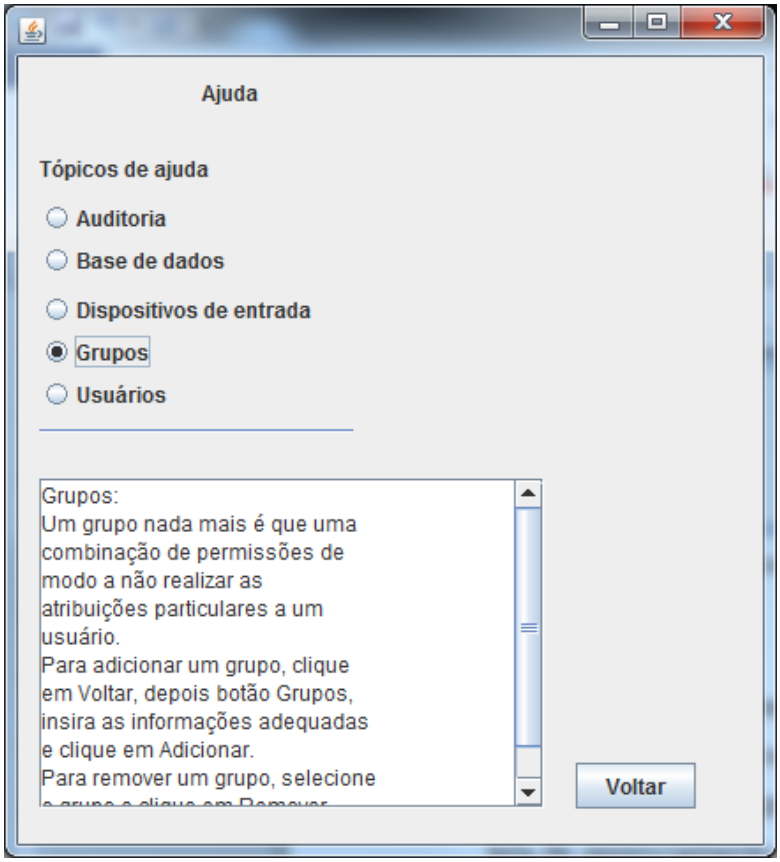

Figura 18 – Tela de ajuda do sistema

4 TESTES E VALIDAÇÃO

Este capítulo descreve os testes realizados no sistema e seus resultados. Os testes foram selecionados para verificar o desempenho do sistema em diferentes aspectos.

4.1 Teste de aceitação do *polling*

Este teste visou verificar a quantidade de tempo em que o dispositivo de entrada continuou funcionando apenas com solicitações de leitura de seu servidor *web*. Foram realizadas oito sessões de quatro horas, com apenas um travamento em virtude de uma queda de luz no CTISM.

4.2 Teste de estresse

Este teste consiste em inserir um objeto (no caso, uma borracha) entre os botões, de modo a forçar a inserção de dados, por quatro horas, durante oito sessões. Foram constatados em média 500 acessos confirmados com abertura de porta antes que o dispositivo PIC *Mini Web* travasse. Após esse momento, o dispositivo somente retornou a funcionar retirando-o da energia e religando. O resultado deste teste pareceu bastante promissor, pois parece improvável que em um ambiente escolar entre essa quantidade de pessoas em um período de quatro horas.

4.3 Teste de senha errada

Em virtude da programação realizada no teclado, as respostas erradas são ignoradas, caindo no *timeout* que existe após 5 segundos sem respostas. Por esse motivo, foi necessário testar essa característica. Também foram realizadas oito sessões de 4 horas. O resultado desse teste foi uma média de 400 tentativas antes do travamento.

4.4 Teste de senha correta, porém sem permissão na porta

Esse caso é o mesmo do anterior, pois as respostas erradas são ignoradas. Por via das dúvidas, foi testado também durante oito dias, quatro horas por sessão. A média foi a mesma anterior, 400 tentativas antes do travamento.

4.5 Simulação do ambiente real

Por não ser possível adicionar o dispositivo à porta em tempo hábil, em virtude da falta da fechadura eletrônica ter sido disponibilizado apenas na ultima semana da implementação do projeto, foi inserida uma regra na sala. Cada vez que algum usuário entrava na sala, era obrigado a digitar suas credenciais no dispositivo de entrada. Ao final de uma semana, o dispositivo travou aproximadamente três vezes. Acreditamos que para um ambiente acadêmico este resultado é satisfatório.

4.6 Resultados

Apesar dos testes positivos, deve ser observada a limitação computacional dos recursos que esse dispositivo em particular possui. Mesmo que possua um servidor *web,* infelizmente esse servidor aceita poucas conexões, tornando talvez inviável por não suportar a sobrecarga de solicitações quando essas tiverem um tempo pequeno entre as consultas, pois o PIC Mini Web poderia se confundir e travar, como aconteceu em alguns momentos. Caso isso aconteça em um ambiente real torna-se incômodo desligar e ligar o aparelho apenas para poder atravessar uma porta.

Considerando um dispositivo com melhor desempenho aliado ao sistema desenvolvido, é possível desenvolver uma solução eficiente o bastante para a utilização em larga escala no CTISM, porém, este dispositivo atual deixa muito a desejar.

5 CONCLUSÃO E TRABALHOS FUTUROS

Ao fim desse trabalho, foi constatado que é possível desenvolver um *software* experimental para controle de acesso que possui tipos distintos de dispositivos de entrada, incluindo dispositivos de baixo custo. Também foi possível notar a importância dos recursos computacionais, quando se tratando de dispositivos embarcados, uma vez que atualmente os computadores estão cada vez mais potentes. A dificuldade em utilizar um dispositivo com recursos computacionais tão limitados instiga a executar uma programação mais robusta e que desperdice menos recursos, exigindo muito mais do programador.

Enquanto o projeto era realizado e implementado, mais requisitos não funcionais e mais desafios inesperados surgiam, obrigando a limitação de escopo para que o projeto fosse concluído com êxito no prazo estabelecido.

A primeira consequência deste trabalho foi o aumento da habilidade de programação, pois os desafios impostos obrigaram a sessões de pesquisa sobre assuntos novos, muito associados à tecnologia atual.

A segunda consequência foi constatar que esse sistema pode entregar o que foi proposto, ou seja, ajudar a aumentar a segurança no acesso aos locais restritos do CTISM.

Por fim, um dos projetos futuros é aumentar as funcionalidades desse trabalho, fazendo com que toda a administração do sistema seja realizada via *web,* a adição de dispositivos de entrada e bases de dados via *plug-ins,* utilizando dispositivos embarcados com mais recursos computacionais.

REFERÊNCIAS BIBLIOGRÁFICAS

ABNT. ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **NBR ISO/IEC 17799**: Tecnologia da Informação. Código de Prática para Gestão da Segurança da Informação. Rio de Janeiro, 2003.

BARKLEY J. et al. Role Based Access Control for the World Wide Web, **National Institute of Standards and Technology**. Maryland, EUA, p. 2, Abril 1997. Disponível em: <http://csrc.nist.gov/groups/SNS/rbac/documents/web_servers/barkley-et-al-97.pdf>. Acesso em 25/10/2011.

CHESTWICK, W. R., BELLOVIN, S. M., RUBIN, A. D. **Firewalls e segurança na Internet – Repelindo o hacker ardiloso.** 2ª ed. Porto Alegre : Bookman, 2005.

METSKER, J. S., WAKE, W. C. **Design Patterns in Java.** 1ª ed. Boston, United States: Pearson Education, Inc, 2006.

FLANAGAN, D. **Java in a nutshell**, A nutshell handbook in a Nutshell. 5ª ed. Cambridge, United States: O'Reilly Media, Inc, 2011.

IBM, **Glossário,** Política de controle de acesso. Disponível em: <http://publib.boulder.ibm.com/infocenter/wchelp/v5r6/index.jsp?topic=/com.ibm.commerce.admin.doc/concepts/caxaccesspolicy.htm>. Acesso em 11/12/2011.

ID TECH, **Soluções**. 2006. Disponível em: <http://www.idtech.com.br/solucoes.asp>. Acesso em 08/11/2011.

JORDAN, S. C. **A Guide to Understanding Discretionary Access Control in Trusted Systems.** Maryland, United States: DIANE Publishing, 1987.

Hailperin, M. **Operating systems and middleware:** supporting controlled interaction. United States: Course Technology, 2007.

MADIS, **Soluções.** 2011. São Paulo. Disponível em: <http://www.madis.com.br/?t=produto&cat=2>. Acesso em 08/11/2011.

MATTOS, C. L. A. **Sentinel:** um engenho Java para controle de acesso RBAC. 2003. Pernambuco. Disponível em: <http://www.cin.ufpe.br/~tg/2003-1/clam.doc>. Acesso em: 25/10/2011. Trabalho de Graduação em Segurança da Informação. 50 p.

NIBTEC, **Controle de Acesso.** 2011. Minas Gerais. Disponível em: <http://nibtec.com.br/produtos.html>. Acesso em 08/11/2011.

ROCHA, C. C. **Desenvolvimento de uma interface web para integração e configuração da rede de sensores-atuadores do projeto CONVERGE UFSM.** 2007. Santa Maria. Disponível em: <http://www-app.inf.ufsm.br/bdtg/tg.php?id=275>. Acesso em: 09/12/2011. Trabalho de graduação em Ciência da Computação. 47 p.

SANTIN, A. O. **Teias de Federações:** uma abordagem baseada em cadeias de confiança para autenticação, autorização e navegação em sistemas de larga escala. Florianópolis. 2004. Disponível em: <http://www.ppgia.pucpr.br/~santin/ftp/tese/TEIAS%20DE%20FEDERACOES%20-%20SANTIN.pdf>. Acesso em 02/12/2011. Tese de Doutorado em Engenharia Elétrica, Área de Sistemas de Informação. 179 p.

SANTOS, A. **Gerenciamento de Identidades.** Rio de Janeiro: Brasports, 2007.

_____. **Quem mexeu no meu sistema?** Rio de Janeiro: Brasports, 2008.

SIEMENS, **Controle de Acesso Remoto.** 2011. Disponível em: <http://www.industry.siemens.com.br/buildingtechnologies/br/pt/seguranca-eletronica/monitoramento-24h/controle-acesso-remoto/Pages/controle-acesso-remoto.aspx>. Acesso em 08/11/2011.

VASLYN, R. et al. Trusted computing – A new challenge for embedded systems. **International Conference on Electronics, Circuits and Systems 2006.** Nice, France. 2006. Disponível em: <http://hal.archives-ouvertes.fr/docs/00/12/44/15/PDF/icecs06_vaslin.pdf> Acesso em 04/11/2011.