Universidade de São Paulo

Escola Politécnica

Engenharia da Computação Cooperativo

FERNANDO HENRIQUE GINES

THIAGO TADEU TSAI

Projeto e Implementação de um sistema de identificação por RFId para uma aplicação de automação residencial

São Paulo

2007

FERNANDO HENRIQUE GINES
THIAGO TADEU TSAI

Projeto e Implementação de um sistema de identificação por RFId para uma aplicação de automação residencial

Trabalho apresentado como requisito para obtenção do grau de Engenheiros com ênfase em Computação pela Escola Politécnica da Universidade de São Paulo.

São Paulo
2007

FERNANDO HENRIQUE GINES

THIAGO TADEU TSAI

Projeto e Implementação de um sistema de identificação por RFId para uma aplicação de automação residencial

Trabalho apresentado como requisito para obtenção do grau de Engenheiros com ênfase em Computação pela Escola Politécnica da Universidade de São Paulo.

ORIENTADOR: PROFA. DRA. TEREZA CRISTINA CARVALHO

São Paulo

2007

Aos meus pais, irmãos, sobrinhos e amigos.

Fernando Ginês

Ao meu pai que sempre foi um exemplo de determinação e dedicação à família, à minha mãe que me ensinou os bons princípios da alma e do coração, aos meus irmãos pelo apoio e amizade e a tantos amigos sem os quais não teria conseguido.

Thiago Tsai

AGRADECIMENTOS

Aos nossos pais, por toda a atenção e apoio que sempre nos deram até hoje.

Aos familiares e amigos, que acompanharam e torceram por nós ao longo dessa longa jornada.

A Priscila Cardoso Ferreira, que nos colocou em contato com o Projeto *Autonomamente*. Ao Prof. Thimoty Barbieri que nos orientou durante nosso percurso na Itália e aos demais professores que participaram, junto com a associação So.La.Re, do projeto *Autonomamente*.

À orientadora Profa. Dra. Tereza Cristina Carvalho, pela orientação na elaboração deste trabalho.

ABSTRACT

In the present thesis we propose a framework for continuous and transparent human identification of multiple users using the RFId technology. In our framework, users wearing a wristband are automatically detected when they approximate to the user's terminal equipped with an RFId reader. This happens through a continuous polling identification system, which detects the tag and identifies the user. After that, the framework warns the domotic application, which will perform the suitable tasks. The main goal of the system is to easily identify users with different backgrounds and skills by using a ubiquitous solution and thus enhancing the degree of usability on domotic applications. Our work is inserted on a project of domotic homes called Autonomamente, developed through a collaboration agreement between the Politecnico di Milano (Technical University of Milan) and the Association So.la.re. Inside the house, users will interact with an embedded system, which is dedicated to aid disabled people on daily activities. Following the role inside the Autonomamente project, our work consisted of two parts. The first was to provide an identification framework responsible for taking care of RFId complexities and providing transparent functionalities for the application. The second was to implement a Home Automation Module responsible for communicating with the domotic bus and control house devices from an Adobe Flash application. We started from an analysis of the RFId technology, which gave us the knowledge needed to develop an RFId-based application. The choice of the RFId equipment was constrained by the low cost solution aspired by the Autonomamente project. After the design and implementation on a test prototype, some tests were performed aiming to verify the performance and behavior of the system as a pervasive identification solution. The framework achieved satisfactory results on all criteria with exception to the distance to the detected tags. This problem was bound to the acquired RFId equipment and could be solved using a RFID reader with longer reading range. Performance tests were also performed in the Home Automation Module and, in this case, all criteria achieved satisfactory levels with no exception.

RESUMO

No presente trabalho nós propomos um *framework* para identificação humana pervasiva[1] de múltiplos usuários através da tecnologia de RFId (*Radio Frequency Identification*). Em nosso sistema, usuários usando uma pulseira são automaticamente identificados ao se aproximarem do terminal de usuário. Isto acontece por meio de um sistema de identificação por varredura (polling) que detecta a etiqueta (*tag)* RFId embutido na pulseira e identifica o usuário. Após isso, o *framework* avisa a aplicação de automação residencial sobre eventos de aproximação ou afastamento do usuário e esta, por sua vez, efetua as tarefas como melhor convém a aplicação. O principal objetivo do sistema é facilitar a identificação de usuários possuidores de diferentes perfis e habilidades através de uma solução pervasiva e, portanto, aumentando o grau de usabilidade de aplicações de automação residencial. Este projeto está inserido em um projeto Italiano de lares automatizados chamado Autonomamente, objeto de colaboração entre a Politecnico di Milano (Technical University of Milan) e a Associação So.la.re para deficientes e idosos. No interior do lar automatizado, pessoas com deficiência interagem com um sistema embarcado dedicado para auxiliar pessoas deficientes a terem uma vida mais autônoma. Através de uma interface adaptável às deficiências do usuário, o sistema disponibiliza novas possibilidades para a comunicação, para o controle de dispositivos da casa e identifica ameaças à segurança (*safety*) do lar. Essas funcionalidades foram desenhadas respeitando os requisitos de usabilidade e comunicação adequados a essa categoria de usuários. Nosso papel dentro do Projeto Autonomamente consistiu de duas partes: 1. Disponibilizar um *framework* para identificação humana responsável por cuidar de todas as complexidades do equipamento RFId e fornecer funcionalidades transparentes às aplicações de automação residencial. Através destas funcionalidades, outros módulos dentro da aplicação são capazes de controlar o sistema de identificação e de verificar a

---

[1] O termo pervasividade é usado em computação para interações homem máquina sem que o homem precise necessariamente estar consciente da interação.

presença de um determinado usuário sob demanda a fim de autenticar as suas operações durante a utilização do sistema; 2. Implementar o módulo de Automação Residencial responsável, de maneira análoga, por fornecer as funcionalidades para comunicação e controle dos dispositivos residenciais através de uma interface feita em Adobe Flash. Iniciou-se o trabalho com a análise da tecnologia RFId, o que forneceu o conhecimento necessário para desenvolver uma aplicação baseada em RFId. A escolha do equipamento RFId foi restringida a opções de baixo custo aspiradas pelo projeto Autonomamente. Depois de desenhar e implementar a solução em um protótipo de *hardware*, alguns testes foram efetuados visando verificar o desempenho e o comportamento do sistema segundo a perspectiva de uma solução pervasiva. O *framework* de identificação humana atingiu resultados satisfatórios em todos os critérios com exceção da distância de detecção do usuário. Este problema está relacionado ao equipamento RFId adquirido e poderia ser resolvido com a aquisição de um leitor RFId de maior raio de leitura. Testes de desempenho também foram efetuados no módulo de automação e, neste caso, todos os critérios atingiram níveis satisfatórios sem exceção.

SUMÁRIO

LISTA DE TABELAS

LISTA DE FIGURAS

LISTA DE ABREVIATURAS E SIGLAS

Adobe Flash	Plataforma de desenvolvimento de conteúdo multimídia interativo para a Internet
BTicino	Empresa multinacional no ramo de infra-estrutura elétrica de construções
EEPROM	*Electrically Erasable Programmable Read-Only Memory*
GUI	*Graphic User Interface*
ISO	*International Organization for Standardization*
ISTAT	Instituto Italiano de Estatísticas
LF	*Low Frequency*
OpenWebNet	Protocolo usado para comunicação de componentes de automação de construções da BTicino
RFId	*Radio Frequency Identification*
ROM	*Ready Only Memory*
RS232	Padrão de comunicação de dados também conhecido como Serial.
Sistemas *domóticos*	Tecnologias no campo de automação residencial
Text to Speech	Funcionalidades em que o computador pronuncia o texto de forma audível
UHF	*Ultra High Frequency*
UML 2.0	Versão 2.0 da linguagem de modelagem de sistemas *Unified Modeling Language*
Verichip	Tecnologia de tag RFId usada para implantes dentro de seres humanos
XML	*Extended Markup Language*

1 INTRODUÇÃO

Este documento apresenta a motivação, os objetivos perseguidos, a concepção da solução e o processo de desenvolvimento do sistema de identificação humana pervasiva através da tecnologia RFId e do sistema para controle da *aplicação domótica*[2] através de interfaces adaptáveis[3].

1.1 MOTIVAÇÃO

Nos últimos anos, a computação pervasiva tem crescido em popularidade. Os recentes avanços da computação e sua integração nas atividades diárias aumentam o desejo por interações mais rápidas e menos intrusivas com os recursos computacionais. Neste contexto, parte do interesse de pesquisa está focado no estudo de ambientes inteligentes e sistemas *domóticos*, onde soluções estabelecidas podem ser facilmente encontradas. Em particular, pessoas com deficiências e idosos podem se beneficiar ainda mais com tais tecnologias, pois estas abrem novas possibilidades para a execução de atividades diárias essenciais, proporcionando uma vida mais autônoma.

Para esta categoria de usuários, a usabilidade tem um papel fundamental e pode determinar até que ponto uma aplicação *domótica* pode ser usada com efetividade, eficiência e satisfação [17]. Métodos de Identificação baseados em RFId (*Radio Frequency Identification*) são uma solução adequada a tal propósito, pois possuem elevado grau de usabilidade, baixa intrusividade, além de uma sinergia tecnológica com sistemas *domóticos*.

---

[2] Aplicação domótica: aplicação no ramo de automação residencial através do qual é possível controlar os dispositivos eletrônicos da construção.

[3] Interfaces adaptáveis: interfaces gráficas que podem mudar dependendo do perfil de usuário.

Este projeto está inserido em um projeto Italiano de lares automatizados chamado "Autonomamente", objeto de colaboração entre a Politecnico di Milano (Technical University of Milan) e a Associação So.la.re para deficientes e idosos. No interior do lar automatizado, pessoas com deficiência interagem com um sistema embarcado dedicado para auxiliar pessoas deficientes a terem uma vida mais autônoma. Através da tecnologia, o sistema disponibiliza novas possibilidades para a comunicação, para o controle de dispositivos da casa e identifica ameaças à segurança (*safety*) do lar. Essas funcionalidades foram desenhadas respeitando os requisitos de usabilidade e comunicação adequados a essa categoria de usuários.

Para identificar usuários com diferentes tipos de conhecimentos e habilidades (físicas e mentais), uma solução pervasiva tornou-se necessária ao projeto. Uma vez corretamente identificados pelo sistema, usuários deficientes podem se beneficiar da aplicação *domótica* e da interface de usuário adaptada às suas capacidades para efetuar tarefas que antes poderiam ser impossíveis ou exigir grande esforço para serem executadas. Um exemplo de tal situação seria a tarefa de verificar as luzes e o fechamento do portão da casa por uma pessoa com deficiência motora ou funcional. O que poderia levar minutos e requerer grande esforço transforma-se em uma simples interação com o sistema.

Aplicações *domóticas* também podem elevar a segurança (*safety*) dos deficientes, sobretudo nos casos de deficiência cognitiva e sensorial. Um vazamento de gás significa alto risco, sobretudo para pessoas incapazes de identificar tal vazamento ou que sofrem de problemas de esquecimento. Neste caso, sensores conectados ao barramento domótico podem cortar imediatamente o fornecimento de gás e comunicar o fato a administradores do sistema *domótico*.

A fim de esclarecer as motivações mencionadas, as próximas subseções apresentam uma explicação mais detalhada sobre a sinergia entre a tecnologia RFId e aplicações *domóticas* e sobre o aumento do grau de usabilidade das aplicações *domóticas* alcançado pelo sistema descrito neste documento:

1.1.1 Aproveitar a sinergia entre a tecnologia RFId e Aplicações *Domóticas*

Métodos de Identificação RFId podem se beneficiar pela sua sinergia com aplicações *domóticas* em relação a outros métodos de identificação humana. Em elevados níveis de automatização, as casas podem ser equipadas com leitores RFId distribuídos, possibilitando identificar usuários, que entram em determinado ambiente, e automaticamente ajustar os dispositivos segundo suas preferências para um horário especificado ou outros critérios predefinidos. Métodos de visualização (*computer vision)* também podem fornecer tal funcionalidade, mas com a desvantagem de trazer a percepção de estar sendo vigiado em sua própria casa.

1.1.2 Aumento do grau de usabilidade de aplicações *domóticas*

Uma vez instaladas em residências e construções, as aplicações *domóticas* podem afetar um grande número de pessoas com as mais diferentes características. Destas, uma atenção especial deveria ser dada aos deficientes e idosos, pois estes podem ser fortemente beneficiados pela tecnologia *domótica*. Ela abre a possibilidade para uma vida mais autônoma ao fornecer novas maneiras de executar atividades diárias. Neste contexto, a usabilidade exerce um papel fundamental e determina a extensão na qual uma aplicação *domótica* pode ser usada por usuários específicos com eficácia, eficiência e satisfação [17].

Pessoas deficientes podem ter 5 tipos de deficiência de acordo com a pesquisa do ISTAT (Instituto Italiano de estatísticas) sobre condições de saúde:

1. Confinamento individual: à cama, ou a uma cadeira - não cadeira de rodas - ou na casa;

2. Deficiência Funcional (por exemplo dificuldade em se vestir, em lavar, em tomar banho, em comer sem ajuda);

3. Deficiência Motora (por exemplo dificuldade em andar, subir escadas, ir para a cama, em se sentar sem ajuda);

4. Deficiência Sensorial (dificuldade para escutar, enxergar, ou falar).

5. Deficiência Cognitiva (dificuldade em tomar decisões considerando uma representação simbólica do mundo).

Para algumas destas características de usuário, não é possível tomar por garantida a hipótese de que o usuário pode sempre escolher corretamente sua identificação no sistema por meio de um procedimento comum de *login/password* ou mesmo pela escolha de sua identificação através de sua própria foto. Este sistema procura identificar usuários com características e habilidades diferentes, usando uma solução pervasiva, aumentando o grau de usabilidade de aplicações *domóticas*.

1.2 OBJETIVO

O trabalho aqui descrito possui dois objetivos principais:

1. Disponibilizar um *framework* de identificação responsável por cuidar de todas as complexidades do equipamento RFId e fornecer funcionalidades de identificação humana para a aplicação *domótica*. Através destas funcionalidades, outros módulos dentro da aplicação devem ser capazes de iniciar ou parar o sistema de identificação e verificar a presença de um determinado usuário sob demanda a fim de autenticar as operações do usuário ao utilizar o sistema.

2. Implementar o módulo de Automação Residencial responsável, de maneira análoga, por prover funcionalidades de comunicação e controle dos dispositivos da casa transparentes ao resto da aplicação, mas também por meio de um sistema de interface adaptáveis como o Adobe Flash.

Em uma visão prática do *framework* proposto, usuários utilizando um bracelete RFId serão automaticamente detectados ao se aproximarem do terminal de usuário equipado com um leitor RFId. A detecção de usuário acontece através de um sistema de varredura (polling) continua que detecta a *tag* RFId contida no bracelete e identifica o usuário. Uma vez identificado, o *framework* avisa os outros módulos da aplicação sobre a presença de um usuário para que estes reajam de acordo com sua especificação. Por meio de um único terminal, o usuário pode controlar os dispositivos da casa.

Para desenhar o sistema de identificação RFId, houveram alguns objetivos relacionados à integração com aplicações *domóticas*, à superação de problemas intrínsecos da tecnologia RFId e à flexibilidade de configuração. Tais objetivos serão detalhados nas subseções seguintes:

1.2.1 Integração do sistema de identificação com aplicações *domóticas*

Este *framework* será construído como um módulo único capaz de tratar todas as complexidades da tecnologia RFId e fornecer funcionalidades transparentes para a aplicação *domótica*. Através do *framework*, outros módulos da aplicação poderão iniciar ou interromper o sistema de identificação, ou verificar a presença de um específico usuário a fim de poder autenticar as operações feitas em seu nome.

1.2.2 Superar o problema de atraso na identificação do usuário

Ao lidar com a tecnologia RFId, uma restrição importante pode afetar a transparência da identificação: o atraso na identificação do usuário. Como um algoritmo de varredura (*polling*) deve ser usado para a detecção das etiquetas (*tags*) RFId, a resposta nunca será imediata. A resposta depende do intervalo de varredura e da capacidade da antena de detectar a etiqueta RFId em um período de tempo curto. Para o usuário, isto significa que sua presença não pôde ser detectada se seus movimentos, ao passarem através do raio de leitura do RFId, forem mais rápidos que o tempo de resposta do sistema.

O sistema requer que uma resposta instantânea seja entregue no momento em que a verificação de presença de um usuário específico é solicitada por outro módulo, conseqüentemente este sistema da identificação deve fornecer os meios para superar o problema tecnológico e fornecer instantaneamente a resposta.

1.2.3 Flexibilidade de configuração

Adicionar, remover e editar configurações de usuário deveriam ser possíveis mesmo após a compilação da aplicação a fim de reduzir o tempo gasto em manutenção. Para tornar isso possível, a configuração do usuário e a relação de etiquetas (*tags*) RFId são fornecidas através de um arquivo XML (*Extended Markup Language*) que pode ser facilmente modificado.

2 ASPECTOS TECNOLÓGICOS E CONCEITUAIS

Neste capítulo, é apresentado um breve resumo sobre as principais tecnologias envolvidas tanto no sistema de identificação contínua de múltiplos usuários como no módulo de Automação Residencial. Iniciamos a seção descrevendo o contexto histórico e técnico da tecnologia RFId e apresentando suas aplicações no campo de identificação humana.

Em seguida, são explicados os conceitos de Sistemas *Domóticos* e Sistemas Embarcados, que juntos compõem o ambiente tecnológico, no qual este projeto está inserido. Finalmente, são apresentados projetos semelhantes a este no meio acadêmico.

2.1 IDENTIFICAÇÃO POR RÁDIO FREQÜÊNCIA (RFID)

RFId é um acrônimo do nome em língua inglesa *Radio Frequency Identification*, que, em português significa Identificação por Rádio Freqüência. Esta seção tem como objetivo fornecer uma visão geral da tecnologia e das implicações de seu uso para a identificação humana.

2.1.1 Breve histórico da tecnologia

O desenvolvimento da tecnologia RFId teve origem na Segunda Guerra Mundial pelo uso da tecnologia de transmissão por rádio freqüência em conjunto com a identificação automática. Sob o comando de Watson-Watt, que chefiou um projeto secreto, os britânicos criaram o primeiro identificador ativo de amigo ou inimigo (IFF, *identify friend or foe*). Foi colocado um transmissor em cada avião britânico. Quando esses transmissores recebiam sinais das estações de radar no solo, começavam a transmitir um sinal de resposta, que identificava o avião como amigo. A tecnologia RFId funciona no mesmo princípio básico. Um sinal é enviado a

um *transponder*[4] o qual é ativado e reflete de volta o sinal (sistema passivo) ou transmite seu próprio sinal (sistema ativo).

Por muitos anos, o desenvolvimento desta tecnologia foi postergado pela inexistência de componentes de pequena dimensão, tais como transistores e circuitos integrados, assim como pela inexistência de microprocessadores. Na década de 50, ocorreu um rápido desenvolvimento nessas áreas, enquanto que a miniaturização e integração ocorreram mais tarde na década de 60.

Enquanto isso, cientistas e acadêmicos dos Estados Unidos, Europa e Japão realizaram pesquisas e apresentaram estudos explicando como a energia RF poderia ser utilizada para identificar objetos remotamente. Pouco depois, no final da década de 60, empresas começaram a comercializar sistemas antifurto que utilizavam ondas de rádio para determinar se um item havia sido pago ou não. As etiquetas de vigilância eletrônica (EAS - *Eletronic Article Surveillance*) utilizam um bit. Se a pessoa paga pela mercadoria, o *bit* é colocado em *off* ou 0e os sensores não dispararão o alarme. Caso contrário, o *bit* continua em *on* ou 1, e se a mercadoria sai através dos sensores, um alarme dispara.

A década de 70 foi caracterizada pelo desenvolvimento da tecnologia eletrônica do RFId. Grandes empresas tais como General Electric, Westinhouse, Philips e Gleynayre começaram a usar a tecnologia RFId para controlar objetos e veículos em movimento. Na Europa, houve um grande desenvolvimento na identificação animal com *transponders* de baixa freqüência, e em Los Alamos com microondas.

A consolidação do RFId como uma tecnologia madura ocorreu somente na década de 80 com o desenvolvimento e difusão global de aplicações nos Estados Unidos, em áreas de controle de mercadorias, meios de transporte, acesso de pessoas e identificação animal. Na Europa, os esforços eram concentrados, principalmente, no uso da tecnologia para identificação animal, atividades industriais e controle de acesso em rodovias.

No final da década de 80, engenheiros da IBM desenvolveram um sistema RFId baseado na tecnologia UHF (*Ultra High Frequency*). O UHF oferece um

---

[4] Transponder: receptor-transmissor que envia um sinal de rádio como resposta para um comando que foi recebido por uma estação remota.

alcance de leitura muito maior e transferência de dados mais velozes. Porém, o custo desta tecnologia era ainda muito alto naquele momento devido ao volume reduzido de vendas e a falta de padrões internacionais. A tecnologia RFId era, portanto, utilizada somente em aplicações complexas, ou aquelas com alto valor agregado que pudessem justificar os altos custos.

A tecnologia RFId moderna chegou somente na década de 90 com a miniaturização dos componentes, reduzindo assim o consumo de energia. Os *transponders* RFId puderam, desta forma, ser alimentados pelo campo eletromagnético do equipamento que o requisita. Além do mais, com a utilização de memória EEPROM (*Electrically Erasable Programmable Read-Only Memory*) ao invés da memória RAM (*Random Access Memory*), não era mais necessário ter uma bateria alimentando constantemente o *transponder* a fim de garantir a permanência dos dados na memória [2].

Outro fator importante que impulsionou a aplicação da tecnologia RFId foi o desenvolvimento de padrões internacionais com o estabelecimento do Auto-ID Center em 1999 no *Massachusetts Institute of Technology* (MIT). Entre 1999 e 2003, o Auto-ID Center ganhou o apoio de mais de 100 grandes empresas do setor de bens de consumo, do Departamento de Defesa dos Estados Unidos e dos principais vendedores de tecnologia RFId.

2.1.2 Arquitetura Clássica da Tecnologia RFId

Para melhor entender a tecnologia RFId é necessário descrever a sua estrutura e o fluxo de informações. Os dispositivos de *hardware* que compõem a arquitetura de sistema do RFId são três (Figura 1):

- *Transponder (Tag)*;
- Leitor;
- Computador *host*.

Sem um desses componentes a tecnologia RFId está incompleta, e é impossível, portanto, acessar as informações contidas nas etiquetas RFId.

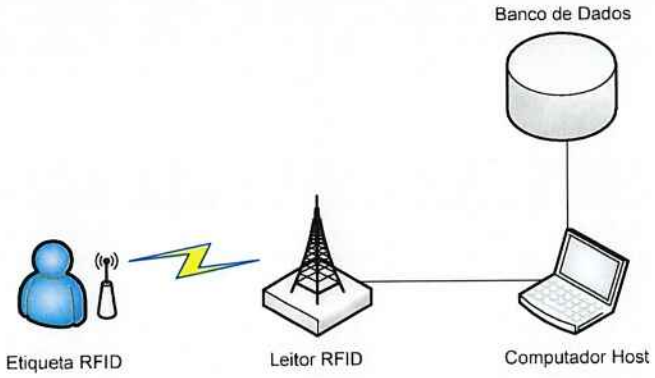

Figura 1 - Arquitetura clássica da tecnologia RFId

Na arquitetura básica da tecnologia RFId, objetos são individualmente equipados com um rótulo (tag) pequeno e barato. Este rótulo (tag) contém um transponder com chip de memória digital, ao qual é atribuído um código eletrônico único de identificação. O interrogador, uma antena embalada com um transmissor e receptor e um decodificador, emite um sinal ativando a etiqueta RFId, podendo ler os dados contidos assim como escrever novos dados. Quando uma etiqueta RFId passa por um campo eletromagnético, detecta o sinal de ativação do leitor. O leitor decifra os dados criptografados nos circuitos integrados da etiqueta e o dado é transmitido para o computador servidor hospedeiro. O software da aplicação no computador anfitrião processa os dados, freqüentemente comparando ou atualizando os dados no banco de dados. Nos próximos itens, os componentes da arquitetura proposta serão detalhados.

2.1.2.1 Etiqueta RFId

A etiqueta RFId, também conhecida por transponder ou rótulo RFId, é o principal elemento do sistema RFId. De acordo com [2], a etiqueta RFId é um receptor-transmissor que envia um sinal de rádio como resposta para um comando que foi recebido por uma estação remota. A resposta pode conter somente o código de identificação da etiqueta RFId ou qualquer outra informação que esteja armazenada em sua memória.

Existem diversas formas de etiquetas, incluindo etiquetas inteligentes, simples adesivos, cartões inteligentes ou chaveiros. No entanto, independente da forma, a etiqueta é composta por três partes: chip, antena e encapsulamento [2].

- Chip: é o componente eletrônico que tem a função de administrar todas as identificações e comunicações. Uma vez interrogado, o chip assegura a exatidão do sinal, respondendo em seguida com a sua informação.
- Antena: tem a função de receber e transmitir as informações, e em alguns casos de etiquetas passivas, tem também a função de fornecer energia para o chip.
- Encapsulamento: é o material ou componente sobre o qual o chip e a antena são embutidos, tendo a função de protegê-los.

2.1.2.2 Leitor RFId

O leitor é a porta de comunicação entre o mundo externo e a etiqueta RFId. O leitor tem a capacidade de interrogar individualmente múltiplas etiquetas, enviar e receber dados, gerenciar o dado de múltiplas etiquetas separadamente e ser a interface com o sistema de informações da empresa.

Normalmente, o leitor é dividido fisicamente em duas partes: antena e unidade de controle.

- A unidade de controle é um microprocessador, que permite gerenciar em tempo real: a interface com as antenas, a requisição das etiquetas que entram no campo magnético das antenas, a colisão entre as diversas respostas vindas das etiquetas e a interface com o sistema de informação.
- As antenas são a interface entre a unidade de controle do leitor e a etiqueta (*tag*). As etiquetas passivas somente são ativadas entrando no campo magnético gerado pela antena, tendo essa, portanto o objetivo adicional de fornecer energia [2].

2.1.2.3 Computador *Host*

O computador *host* deve ser adaptado para ser conectado à unidade de controle e para receber o *software middleware* (mediador). Neste caso, "*middleware*" é um termo genérico usado para descrever um software que se encontra entre o leitor RFId e as aplicações da empresa. É um componente crítico de qualquer

sistema RFId, porque os *middlewares* recebem os dados brutos do leitor – um leitor pode ler até 100 vezes por segundo uma mesma etiqueta – filtrar esses dados e repassar somente os dados úteis para os terminais. O *middleware* executa um papel importantíssimo em levar a informação certa para a aplicação certa no tempo certo [2].

2.1.3 A fonte de energia das etiquetas (*Tags*)

As etiquetas RFIds podem ser divididos em etiquetas passivas, semi-passivas, semi-ativas e ativas.

- **Rótulos (tags) passivos**: não possuem fonte própria de energia. Uma vez inseridas dentro do campo eletromagnético do leitor, a antena do *chip* converte a energia eletromagnética em eletricidade podendo, assim, alimentar o *microchip* da etiqueta. Além disso, a etiqueta é capaz de reenviar à informação armazenada no chip modulando as ondas refletidas pela antena. Devido à sua natureza, esses rótulos apresentam um baixo custo e um curto alcance de leitura com relação às etiquetas ativas [18].
- **Rótulos semi-passivos**: são parecidos com as etiquetas passivas, diferenciando-se por conterem uma pequena bateria utilizada somente para ativar o circuito do *microchip*. Sendo assim, para reenviar o sinal de resposta, estes rótulos ainda dependem da energia provida pelo leitor [18].
- **Rótulos semi-ativos**: São muito parecidos com as etiquetas semi-passivas, porém a bateria ao invés de ser utilizada para ativar o circuito do *microchip* é utilizada para ativar os sensores da memória [7]. A emissão do sinal de resposta ainda depende da energia provida pelo leitor.
- **Rótulos ativos**: são aqueles que, ao invés de utilizarem energia eletromagnética do campo, no qual estão inseridos para serem interrogados, utilizam uma fonte de energia própria. A maioria dos rótulos ativos contém um rádio transmissor e uma bateria, permitindo a transmissão contínua de informação ou sinal caso seja exigido, mesmo que não esteja dentro do alcance do leitor. As etiquetas ativas são capazes também de traçar a evolução de alguns parâmetros no tempo, gravando estes dados na sua memória. Devido a essas características, tais rótulos

são utilizados para rastrear itens custosos em longos percursos e custam mais que as etiquetas passivas [18].

2.1.4 Sistema anti-colisão

Outro dos conceitos técnicos básicos da tecnologia RFId é a anti-colisão. No contexto RFId, anti-colisão refere-se a formas diferentes de evitar que ondas de rádio de um dispositivo interfiram com ondas de rádio de outro dispositivo [19].

Dessa forma, leitores RFId devem utilizar algoritmos de anti-colisão para possibilitar que um único leitor leia mais de um rótulo no campo de leitura. Diferentes sistemas já foram inventados para isolar rótulos individualmente. Por exemplo, quando o leitor reconhece que uma colisão ocorreu, é enviado um sinal especial (*gap pulse*). Uma vez recebido esse sinal, cada etiqueta considera um número escolhido ao acaso para determinar o intervalo de espera antes da emissão desse dado. Uma vez que cada uma tem um intervalo único, as etiquetas enviam seus dados em períodos diferentes.

2.1.5 Alcance e freqüência

A distância que um rótulo pode ser lido varia de acordo com a tecnologia do rótulo, o ambiente em que opera, a freqüência de operação, a dimensão da antena e a posição relativa da antena para a etiqueta.

Ambientes, em que há alta concentração de líquidos ou metais, podem alterar a direção das ondas de rádio, comprometendo a interrogação das etiquetas. Rótulos de baixa freqüência, que têm menor distância de leitura, têm maior capacidade de penetrar nesses materiais e sair sem ser danificada.

Rótulos ativos têm maior distância de leitura que as etiquetas semi-ativas e semi-passivas. Estes, por sua vez, têm maior distância de leitura de rótulos passivos. A presença de baterias, que alimenta os transmissores, permite alcances na ordem de quilômetros para rótulos ativos, enquanto as etiquetas passivas alcançam no máximo 10 metros.

A tabela a seguir, fonte de dados [20], mostra a relação entre as freqüências e a distância de leitura, sua interação com a água e metais, a dimensão da etiqueta, fonte de energia e alguns exemplos de aplicação.

Freqüência	Baixa Freqüência 125 - 134,2 KHz	Alta Freqüência 13,56MHz	Freqüência Ultra Alta 860 - 960 MHz	Microondas 2,45GHz e 5,8GHz
Alcance de leitura (rótulos passivos)	0,5m	1,0 - 1,5m	3,0m	5,0 - 10,0m
Capacidade de leitura (metais e líquidos)	Excelente	Boa	Média	Baixa
Dimensão da etiqueta	Muito grande	Grande	Média	Pequena
Fonte de energia	Passiva, indução eletromagnética	Passiva, indução eletromagnética	Ativa com bateria integrada	Ativa com bateria integrada
Aplicações	Monitamento de animais e controle de acesso	Smart Cards, monitoramento de produtos e controle de acesso	Monitoramento de containers, sistemas de transporte	Cadeia de suprimento, sistemas de transporte

Tabela 1 - Características das diferentes freqüências de rótulos RFId

2.1.6 Padronização Internacional

A tecnologia de RFId tem se difundido rapidamente através dos mais variados mercados. A necessidade trocar e compartilhar informação relacionada a objetos ou a pessoas torna necessário o desenvolvimento de padrões internacionais.

Os padrões são importantes para todos quando especificam as exigências de produtos, serviços, processos, materiais, sistemas, e práticas gerenciais e organizacionais, sendo projetados para serem implementados em nível mundial. Os padrões reduzem as barreiras à difusão da inovação tecnológica e facilitam a interação entre equipamentos.

2.1.7 Exposição humana à tecnologia RFId

A exposição humana aos campos eletromagnéticos é também uma preocupação atual para a tecnologia RFId. Nos últimos anos, na Europa uma nova normativa foi liberada para disciplinar a emissão de campos eletromagnéticos nos

mais variados equipamentos. Muitas pesquisas têm sido desenvolvidas em diversos países sobre o risco da exposição humana aos campos eletromagnéticos de freqüência de rádio. Entretanto, os resultados até esse momento não são conclusivos.

Desta maneira, é vital assegurar-se de que os níveis da emissão do equipamento de RFId sejam mensuráveis para garantir a conformidade com os regulamentos. Toda a aplicação deve ser prudente e respeitar os limites para a exposição humana aos campos eletromagnéticos, não somente por obrigação moral, mas também para redução de suspeitas de males causados pela tecnologia.

O aspecto da exposição humana à tecnologia de RFId é focado, principalmente, nas aplicações de longo alcance, onde as forças de radiação são mais importantes. De acordo com a "*World Health Organization*" - (WHO) e - "*Ionizing Radiation Protection*" (ICNIRP), a taxa específica de absorção dos sistemas RFId de longo alcance projetados de acordo com os regulamentos existentes é de muitos dB abaixo dos valores máximos considerados perigosos [9].

2.1.8 Restrições para a detecção da etiqueta RFId

Uma restrição da detecção da etiqueta RFId relevante no contexto deste projeto é orientação da etiqueta no campo magnético do leitor. Em particular para a tecnologia de alta freqüência (13,56MHz) utilizada no projeto, a orientação é essencial para que a detecção ocorra com sucesso. A parcela do campo que aciona a resposta é a que passa através da espiral da antena da etiqueta. Conseqüentemente, uma etiqueta de alta freqüência posicionada em paralelo às linhas do fluxo magnético nunca será detectada. Uma inclinação de 45 graus em relação ao ângulo ideal (90 graus) já pode comprometer a funcionalidade da etiqueta. Este problema pode ser solucionado usando um leitor de gerador de ondas circulares polarizadas, ou pelo uso de uma etiqueta (*tag*) com antenas tripolares que sempre respondem para pelo menos uma das direções do campo do leitor [20].

2.1.9 Uso da tecnologia para a identificação humana

Ambos os setores privados e públicos estão usando cada vez mais a tecnologia RFId para monitorar e identificar materiais (como para a gerência de inventário) e pessoas. Na Itália, uma clara predominância de aplicações para a identificação humana em respeito às aplicações para identificação de materiais pode ser observada. De acordo com o relatório da *School of Management of Politecnico di Milano*, comparando aplicações da identificação de seres humanos e de material, 284 aplicações (que representam 63% do total) são apontadas para a identificação humana [20].

Abaixo são mencionados alguns usos atuais da tecnologia de RFId que foca na identificação humana:

- Em São Paulo, os sistemas para pagamento de transporte público estão empregando a tecnologia RFId. A etiqueta RFId é embarcada em uma espécie de cartão de crédito (chamado bilhete único), quando a varredura é feita no cartão, a extração de detalhes sobre o saldo do cartão e a cobrança da passagem podem ser efetuadas.
- Em agosto 2004, o Ohio *Department of Rehabilitation and Correction* (ODRH) aprovaram um contrato $415.000 para avaliar a tecnologia de monitoramento de pessoal usando RFId. Os internos usarão etiquetas RFId embarcadas em pulseiras que podem monitorar os prisioneiros e detectar caso eles tentem remover as pulseiras emitindo um alerta aos computadores da prisão. Este projeto não é o primeiro na tentativa de monitorar prisioneiros nos EUA. As prisões em *Michigan*, na *Califórnia* e em *Illinois* empregam já a tecnologia.
- Em 2004, o escritório do General mexicano Attorney implantou 18 etiquetas RFId nas mãos de membros da equipe de funcionários com o Verichip para controlar o acesso à sala de dados de alta segurança.
- Em outubro 2004, o FDA (*Food and Drugs Administration*) aprovou os primeiros chips RFId que podem ser implantados em seres humanos. Os *chips* de 134 KHz, da VeriChip Corp., podem incorporar a informação médica pessoal e podem salvar vidas e reduzir danos vindos de erros em tratamentos médicos, de acordo com a companhia.

Entretanto, monitorar e identificar seres humanos usando RFId, especialmente quando feito pelo governo, apresentam ameaças sérias à privacidade. Peritos de segurança advertem sobre o uso do RFId para a autenticação de pessoas devido ao risco de roubo de identidade. Fraudes seriam possíveis de maneira que uma pessoa possa roubar a identidade de outra em tempo real. Devido aos recursos limitados da tecnologia RFId, é virtualmente impossível proteger-se de tais ataques, pois seriam necessários protocolos complexos e sistemas interligados de comunicação à distância.

2.2 SISTEMAS DOMÓTICOS

A palavra *domótica* vem do latin *domus* que significa casa e da palavra robótica. Os sistemas *domóticos* são as tecnologias dentro do campo da automatização residencial especializado para suprir as necessidades para a automatização residências privadas, focadas principalmente na integração de dispositivos eletrônicos, tais como eletrodomésticos e dispositivos de controle, tornando estes edifícios mais "inteligentes".

Os sistemas *domóticos* trazem aos usuários a possibilidade de controlar diversos dispositivos eletrônicos dentro das construções através de uma única interface comum (Exemplo: *Web Browser*, Telefone, *Touch Screen*). Por ser baseado em imagens, esse tipo de interface é muito mais intuitiva se comparada às interfaces normalmente disponíveis em construções. O esquema (Figura 2) mostra a arquitetura principal de um sistema domótico, nela é possível conectar sensores, atuadores, dispositivos eletrônicos e equipamentos de audio/video no barramento domótico. Aplicações *domóticas* (provedoras das interfaces adaptáveis) podem controlar os dispositivos através do *Domotic Gateway* que está conectado ao barramento e possui a capacidade de receber e enviar mensagens no protocolo adequado.

Figura 2 - Exemplo de sistema domótico

O escopo no qual as aplicações *domóticas* podem ser aplicadas pode ser:

Aumento do nível de conforto: automatização de interruptores, ignições e todos os pontos de iluminação da casa, a criação e o uso de esquemas apropriados para circunstâncias específicas (como jantar, leitura ou dormir) e controle remoto de dispositivos eletrônicos (como um forno, microondas e uma máquina da lavagem);

Aumento da segurança: a televisão de circuito fechado, sensores de presença, vídeo interfone integrado com reconhecimento facial que pode detectar pessoas conhecidas ou intrusas;

Economia de energia: o controle da quantidade da iluminação, da temperatura do ambiente e do interrompimento automático de dispositivos elétricos pode conservar uma quantidade considerável de energia. A automatização pode também coletar dados úteis para o melhor uso da energia e melhorar o balanço de consumo.

Todos estes escopos tentam reduzir a necessidade da interação humana com os dispositivos eletrônicos, economizando tempo e reduzindo a possibilidade de erros. Desta forma, obtêm-se uma melhora na qualidade de vida do usuário.

A comunicação entre os dispositivos eletrônicos e o sistema de controle deve usar um protocolo comum para que os componentes possam interoperar. Muitos protocolos podem ser usados para esta interação, como o X10, o EIB, o EHS, o OpenWebNet e o Konnex. Este último, Konnex, é um protocolo que engloba três outros protocolos: o X10, o EIB e o EHS. A presença de um único padrão que garanta a interoperabilidade entre os produtos dá aos colaboradores e aos construtores um grau elevado de flexibilidade para estender e modificar instalações eletrônicas.

A tecnologia *My Home* é uma solução desenvolvida pela *Bticino* usando o protocolo *OpenWebNet*. Ela dispõe de uma quantidade extensa de dispositivos que

podem controlar quase todas as funcionalidades dentro e fora da construção. A tecnologia *My Home* foi projetada para permitir que o usuário controle sua casa não apenas localmente, mas também remotamente, através da Internet. Ela provê ao usuário o nível máximo de conforto e de qualidade de vida dentro de seu próprio lar.

O protocolo de Konnex é o padrão europeu, criado pela associação KNX, de redes de dispositivos eletrônicos para construção. A associação KNX tem mais de 90 membros, sobretudo fabricantes de produtos para automação residencial e construções, e faz parceria com mais de 40 universidades ou escolas técnicas a nível mundial.

2.2.1 Sistemas *domóticos* para pessoas deficientes ou idosas

Os sistemas *domóticos* trazem a possibilidade de ajudar pessoas que têm algum tipo de dificuldade em executar atividades diárias essenciais, como idosos e deficientes. Estas pessoas podem ser ajudadas por tecnologias que facilitem a execução de atividades cotidianas sem o auxílio direto de outros seres humanos, tal auxílio pode ocorrer, por exemplo, por meio de residências automatizadas. Para usuários deficientes, os sistemas *domóticos* podem oferecer interfaces que centralizam o controle dos dispositivos e facilitam a utilização se comparados com os sistemas de dispositivos disponíveis em construções normais.

Integrados com o sistema domótico, programas de comunicação como telefone, e-mail e o *web browser* podem também ser adaptados de acordo com as necessidades do usuário, oferecendo a pessoas que não poderiam usar tais programas sozinhos uma nova maneira fazê-lo.

Na Itália um número elevado de conferências e exibições sobre este assunto tem sido organizado nos últimos anos, tais como "Handimatica" [21] e "Ability" [22], mostrando o interesse crescente na utilização de sistemas *domóticos* para o auxílio a pessoas deficientes e idosas.

2.2.2 Protocolo OpenWebNet

Este item tem como objetivo fornecer uma visão geral do funcionamento do protocolo utilizado no módulo de Automação Residencial apresentado neste trabalho.

Uma mensagem do protocolo OpenWebNet (mensagem OPEN) consiste de uma seqüência de caracteres do conjunto: (0,1,2,3,4,5,6,7,8,9 ,*,#). Uma mensagem tem a estrutura conforme o esquema abaixo:

* campo1 * campo2 * campo3 * campo4 * campo5 ... * campoN # #

A mensagem sempre começa com o caracter '*' e sempre termina com '##'. Os caracteres ' * ' são usados como um separador de campos. Existem cinco tipos de campos: WHO (quem), WHAT (o que), WHERE (onde), DIMENSION (dimensão) e VALUE (valor). As possíveis mensagens são listadas na próxima tabela com sua sintaxe:

Messagem	Sintaxe
ACK	*#*1##
NACK	*#*0##
Padrão	*WHO*WHAT*WHERE##
Requisição de Estado	*#WHO*WHERE##
Requisição com Dimensão	*#WHO*WHERE*DIMENSION##
Escrita com Dimensão	*#WHO*WHERE*#DIMENSION*VAL1*VAL2*..*VALn##

Tabela 2 - Tipos de mensagem OPEN

O campo WHO identifica o serviço ou função do sistema envolvido na mensagem OPEN. A tabela a seguir mostra a associação entre o valor do campo WHO e o tipo de serviço.

Valor do WHO	Serviço
0	Cenários
1	Iluminação
2	Automação
3	Gerenciamento Energético
4	Aquecimento
7	Multimídia
13	Interface com periféricos remotos
16	Sistema Sonoro
1004	Diagnóstico de Temperatura

Tabela 3 - Valores do campo WHO e seu significado

O campo WHAT identifica a ação a ser executada. Existe uma tabela de valores WHAT para cada valor WHO (serviço/função). O campo WHERE identifica o grupo de objetos envolvidos na mensagem, podendo referir-se a um grupo de objetos, a um ambiente específico ou a um único objeto. Tal como acontece com no campo WHAT, existem diferentes campos WHERE permitidos, de acordo com o valor do campo WHO na mensagem.

2.2.2.1 OpenWebNet Sessions

O cliente conecta com o servidor OPEN através de uma sessão TCP/IP. Existem três etapas para criar uma sessão: conexão, identificação e comunicação, ilustrado a seguir (Figura 3):

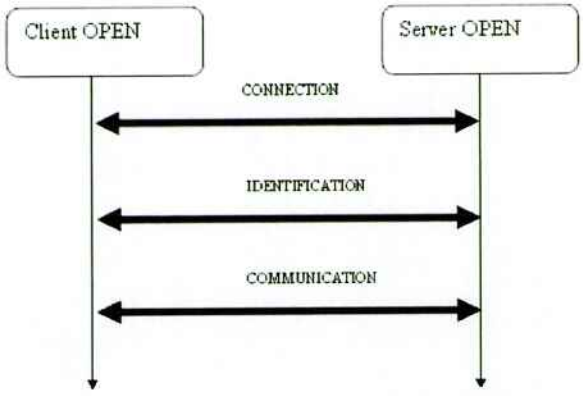

Figura 3 - Comunicação Cliente-Servidor

Durante a primeira fase, o cliente tenta estabelecer uma conexão com o servidor. O servidor irá rejeitar a conexão caso o número de clientes já conectados ultrapassar o número máximo de clientes. Na fase de Identificação, o servidor verifica se o cliente necessita autenticação. O servidor contém uma lista de endereços IP que a conexão é aceita sem pedir ao cliente uma senha. A ilustração abaixo (Figura 4) mostra um exemplo de conexão no modo de comando e no modo de monitoramento:

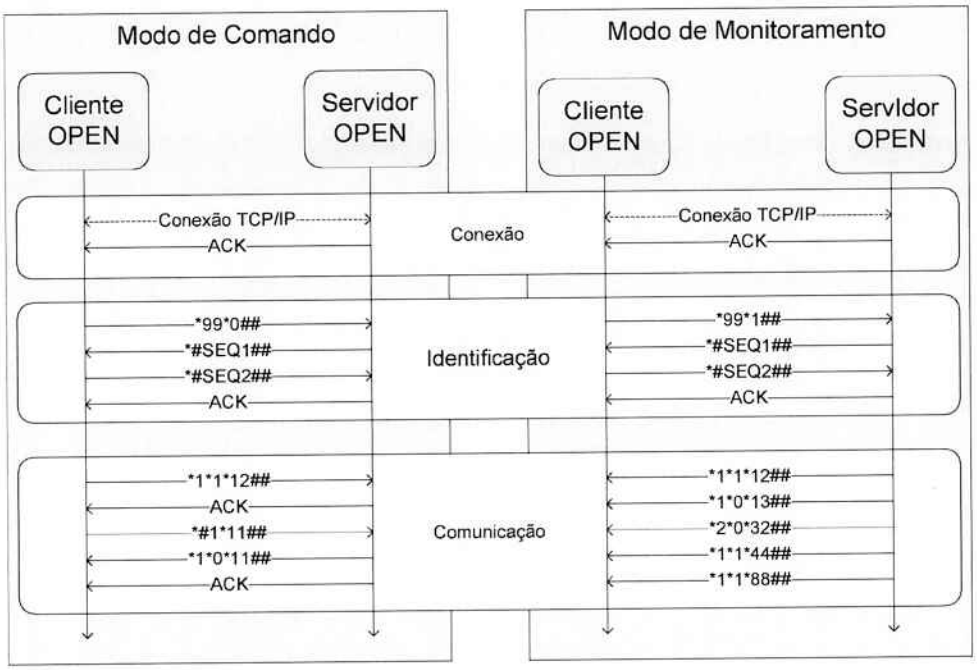

Figura 4 - Exemplo de conexão no modo de comando e no modo de monitoramento.

A fase de Comunicação pode variar de acordo com o tipo de conexão que o cliente quer estabelecer. A ligação pode ser no modo de comando ou no modo de monitoramento. No modo comando, o cliente pode enviar uma mensagem OPEN e aguardar a resposta. No modo de monitoramento o cliente não envia nenhuma mensagem, mas recebe mensagens assíncronas do servidor. Esta última fase dura até o cliente desconectar.

2.3 SISTEMAS EMBARCADOS

O termo sistema embarcado é usado para sistemas computacionais projetados para executar uma função dedicada. Diferentemente dos sistemas de

propósito geral, como os computadores de uso pessoal (*Personal Computers*), sistemas embarcados executam uma ou mais tarefas pré-definidas, normalmente satisfazendo requisitos específicos, e freqüentemente são compostos por uma combinação de componentes de tarefa específica como *hardwares*, *softwares*, e em alguns casos por partes mecânicas não comumente encontradas em computadores de propósito geral [1].

Como estes sistemas são dedicados a tarefas específicas, eles podem ser projetados de maneira otimizada e, conseqüentemente, terão tamanho e/ou custos reduzidos. Sistemas embarcados são freqüentemente produzidos em massa, beneficiando-se de economias de escala. Dos nove bilhões de processadores manufaturados em 2005, menos de 2% tornam-se as unidades de processamento das novas estações *PCs*, *Macs* e *Unix*. Os outros 8,8 bilhões são destinados a sistemas embarcados [1].

São considerados sistemas embarcados dispositivos que vão desde portáteis como *Mp3 players* e câmeras digitais, sistemas de porte médio como impressoras e máquinas fax, até equipamentos de plantas industriais como controladores de fábrica ou sistemas de controle para fontes de energia nuclear. Eles podem executar sistemas simples, como microcontroladores, até sistemas complexos compostos por múltiplas partes, como periféricos e controladores de rede montados dentro de um único e grande chassis. Em alguns casos, os sistemas embarcados são apenas uma parte de um sistema de maiores proporções, como no caso de sistemas anti-roubo para carros.

Alguns dos produtos de sistemas embarcados executam um pequeno programa *assembly* guardados em memória ROM (*Read Only Memory*) sem nenhum sistema operacional; muitos outros executam sistemas operacionais de

tempo real e complexos programas em *multithread* feitos em C ou C++; também está se tornando cada vez mais comum encontrar variantes de sistemas operacionais enxutos derivados do *Linux* e *Windows* controlando dispositivos mais poderosos que ainda assim são considerados sistemas embarcados.

Características comuns:

Algumas características dos sistemas embarcados são:

- Função singular
    - Executam um único programa, repetidamente
- Fortemente restrito
    - Baixo custo, baixo consumo, pequeno, rápido, etc.
- Reativos e de tempo-real
    - Continuamente reagem as mudanças no ambiente do sistema
- Processam certos resultados em tempo real sem atraso
    - Sistemas embarcados podem tanto não ter interface de usuário como possuir interfaces completas similares aos sistemas operacionais em dispositivos como *PDAs*.

Em sistemas mais complexos, uma tela gráfica com sensores de toque (*touch screen*) ou botões ao redor a tela fornecem flexibilidade enquanto minimizam o espaço usado: o significado dos botões pode mudar com a tela, e a escolha envolve o gesto natural de apontar para o que é desejado [14].

O crescimento da internet fornece aos sistemas embarcados outra opção adequada ao controle remoto: uma página *web* que pode ser acessada pela rede. Esta evita o custo de telas sofisticadas nos dispositivos e ao mesmo tempo dispõe de um poderoso sistema de inserção de dados [14].

2.4 REVISÃO DA LITERATURA

Parte do interesse de pesquisa tem recentemente se movido em direção a área de ambientes inteligentes e sistemas *domóticos*. Em [10] é proposto um *gateway* domótico capaz de interagir com diferentes dispositivos em sistemas *domóticos* heterogêneos. Essa solução é mais flexível do que a implementada no trabalho aqui apresentado, ela pode interagir não apenas com a tecnologia da desenvolvida pela empresa *BTicino*, mas também com uma grande variedade de protocolos como o *X10*, *CEBus*, *BatiBus*, *Konnex*, etc. Mas as soluções não são totalmente comparáveis, uma vez que a maior parte do nosso trabalho está relacionado a comunicação do barramento domótico através de uma aplicação *Adobe Flash*. Tal funcionalidade não é contemplada pelo gateway proposto em [10] sem modificações.

Outro trabalho que pode ser adequada a sistemas embarcados que visam ajudar pessoas deficientes é proposto em [16]. Este trabalho propõe, através de um sistema segurança interno, controlar o ambiente pelo comportamento humano, detectando eventos perigosos (com a queda de uma pessoa) e reagindo a esses eventos.

No assunto de sistemas de identificação humana, projetos como o *Gaio* [12], *One World* [5], *Microsoft Easy Living* [4], e o *CORTEX* [15] visam desenvolver uma infraestrutura para dar suporte a ambientes inteligentes de uma maneira abrangente. Eles fornecem abstrações básicas e mecanismos para lidar com a dinâmica e a heterogeneidade de ambientes de computação pervasiva.

Construídos sobre estes mecanismos eles fornecem modelos de aplicações genéricas. Existe, porém, uma distância entre abstrações fornecidas por estes projetos e *frameworks* como o nosso, que dão suporte a aplicações específicas para reconhecimento de seres humanos através de um sistemas contínuo e de múltiplos usuários pela tecnologia RFId. Existe um trabalho [13] também focado em dar suporte a aplicações que usam sistemas de identificação baseados em etiquetas RFId, existindo diversas similaridades com a aplicação aqui proposta, mas o propósito principal é o de controlar objetos dentro do lar, não humanos.

As características de usabilidade dos sistemas de autenticação são analisados em [3], onde métodos baseados em RFId são classificados com alto grau

de usabilidade apenas atrás de métodos de identificação por reconhecimento de voz.

3 ESPECIFICAÇÃO DO PROJETO

Neste capítulo são apresentadas as características funcionais e não funcionais do sistema, diagramas em blocos e o detalhamento da estrutura do projeto, seu *hardware* e seu *software*. No primeiro capítulo foram apresentados os objetivos e motivações para a elaboração do projeto, no segundo foi apresentado o embasamento teórico para o entendimento do problema e da solução proposta. Neste capítulo o problema e a solução são detalhados em maior profundidade.

3.1 DESCRIÇÃO DO PROBLEMA

No cenário em que está inserido nosso projeto, pessoas deficientes vivem dentro de um apartamento equipado com um sistema domótico que pode ser controlado através de um terminal *touch screen (tela sensível ao toque)*. A interface de usuário é projetada para ser flexível de forma a adaptar-se às habilidades e deficiências do usuário. Como exemplo é possível citar o uso do sistema por pessoas com problemas cognitivos, que apresentam dificuldades em entender a linguagem escrita. Neste caso, uma vez identificado o usuário e suas deficiências, a interface é capaz de se adaptar e modificar a forma de comunicação, trocando palavras por símbolos mais intuitivos ao deficiente ou fazendo a leitura das palavras através de funcionalidades *text to speech* (leitura de textos).

Como o sistema lida com pessoas deficientes (incluindo casos de deficiência mental e sensorial), não se pode tomar por garantida a hipótese de que os usuários possam escolher corretamente sua identificação, seja por meio de um processo de *login* e senha ou mesmo, em alguns casos, pela escolha de sua própria foto na tela.

Se o usuário cometer um engano ao escolher sua própria identificação, o sistema se adaptará ao perfil errado e, devido à forte ligação entre o perfil do usuário e a forma de interação com o sistema, ele pode ficar incapacitado de continuar interagindo com o mesmo. É por este motivo que existe a necessidade de identificar corretamente o usuário.

Além disso, dentro de cada apartamento viverão mais de uma pessoa, utilizando-se do sistema em turnos. Cada usuário apresenta um perfil diferente de habilidades e deficiências. Por este motivo torna-se necessário que o sistema identifique continuamente os usuários, a fim de saber, a cada momento, quem está em frente ao terminal utilizando o sistema.

A restrição da tecnologia RFId quanto a orientação das etiquetas RFId em relação à antena pode prejudicar a solução pois o usuário deverá estar consciente da posição de sua pulseira para que a detecção ocorra com sucesso e, portanto, a solução deixará de ser pervasiva. Por este motivo este problema deve ser superado para que a posição da pulseira não influa na detecção do usuário.

Uma vez devidamente identificado e utilizando o sistema por meio da interface adaptada, o usuário deve ser capaz de utilizar o sistema domótico que trará a desejada autonomia e a melhora na qualidade de vida. A tecnologia *Adobe Flash* foi escolhida para interface por sua capacidade em fornecer conteúdos multimídia unidos à capacidade de processamento de dados de plataformas de desenvolvimento. Infelizmente, essa tecnologia é voltada para aplicações web e, portanto, soluções de integração com hardwares como os equipamentos RFId e sistemas *domóticos* não são encontradas, seja em código aberto ou como pacote licenciado. Surge então a necessidade de transferir as funcionalidades de RFId e de sistemas *domóticos* para uma plataforma multimídia feita para web, o *Adobe Flash*.

3.2 ESCOPO DA SOLUÇÃO

Para solucionar o problema de identificação de usuários deficientes em aplicações *domóticas* nós propomos um módulo utilizando a tecnologia de identificação por rádio freqüência (RFId), tal módulo seria responsável por identificar corretamente os usuários quando os mesmos estiverem dentro do raio de detecção do leitor RFId. Para que a identificação ocorra, é necessária a presença de uma antena de rádio freqüência junto ao terminal do usuário, e os usuários deverão estar usando uma pulseira com etiquetas RFId embutidas. Esta pulseira é considerada uma solução *pervasiva*, pois ela pode ser normalmente utilizada na vida cotidiana do usuário.

O módulo de identificação também poderá ser utilizado para autenticar as operações do usuário a fim de garantir que as operações sejam válidas apenas quando este estiver presente em frente ao terminal. Uma vez no sistema, a autorização de operações deve ocorrer somente quando o usuário tomar uma ação que seja relacionada com sua privacidade, como o envio ou a leitura de correio eletrônico. A função de verificação da presença pode ser chamada para autenticar operações, porém, nem todas as interações deverão ser autenticadas. Operações como a seleção de um caractere para escrever no corpo de uma mensagem não necessita a verificação, mas a operação de envio da mensagem deverá ser protegida pelo sistema de autenticação. É de extrema importância que o processo de autenticação através da verificação de presença não cause lentidão no sistema, pois ele estará inserido em muitas operações e pode impactar no desempenho de todo o sistema caso não seja bem projetado.

O sistema também deve ser capaz de identificar múltiplos usuários de maneira instantânea. Desta forma, foi utilizado um sistema de varredura (*polling*) que requisita continuamente ao leitor RFId a leitura da região ao redor da antena. Caso um ou mais usuários sejam detectados o sistema deve receber a informação imediatamente e processá-la avisando os módulos adequados na aplicação *domótica*.

Para que a solução de identificação seja *pervasiva* e não exija nenhuma interação do usuário com o sistema é importante que o problema de orientação da etiqueta (*tag*) RFId em relação à antena não influa no processo de identificação.

A utilização de sistemas *domóticos* através de interfaces em *Adobe Flash* deve respeitar os protocolos de comunicação e contemplar todas as funcionalidades da tecnologia *domótica* escolhida, de maneira a funcionar não apenas para os dispositivos disponíveis no protótipo, mas também por todos aqueles compatíveis com o padrão da tecnologia.

3.3 REQUISITOS FUNCIONAIS

Com base na descrição do sistema, especialmente na descrição de seus serviços, é possível listar seus requisitos funcionais. Estes guiam todos os passos conseguintes de desenvolvimento do sistema e de seus testes.

3.3.1 Sistema de Identificação humana através de etiquetas RFId

a) Permitir a inicialização da rotina de detecção de usuários através de outros módulos da aplicação *domótica*.
b) Permitir a paralisação da rotina de detecção de usuários através de outros módulos da aplicação *domótica*.
c) Permitir a verificação de presença física de qualquer usuário do sistema sob demanda de outros módulos da aplicação *domótica*.
d) Assegurar a correta identificação do usuário uma vez que sua pulseira for detectada pelo equipamento RFId.
e) Permitir a detecção do usuário independente da posição na qual pulseira se encontra em seu pulso.

3.3.2 Sistema Domótico

a) Permitir o controle e leitura de um dispositivo eletrônico residencial específico conectado ao barramento domótico sob demanda de uma interface feita em *Adobe Flash*.
b) Permitir o controle e leitura de um conjunto de dispositivos eletrônicos conectados ao barramento domótico sob demanda de uma interface feita em *Adobe Flash*.

3.4 REQUISITOS NÃO-FUNCIONAIS

Como a solução proposta utiliza tecnologias desenhadas inicialmente para propósitos distintos da solução almejada, os requisitos não funcionais são de grande importância e servem para garantir que, apesar das restrições de tecnologia, os níveis de serviço da solução sejam garantidos. Abaixo são retratados os requisitos não-funcionais dos sistemas de identificação e de automação residencial.

3.4.1 Desempenho

A verificação de presença física do usuário, sob demanda de outros módulos, deve ocorrer com baixa latência de modo que seja percebida como instantânea quando ligada a uma ação do usuário. Esta sensação coincide com tempo de resposta inferior a 100ms [6].

3.4.2 Confiabilidade

Relacionado a confiabilidade do sistema é possível citar a importância de que os dados de identificação do usuário passados através dos equipamentos de RFId até o computador host sejam verificados antes de processados a fim de garantir a correta identificação. Este requisito é especialmente importante uma vez que a tecnologia por rádio freqüência está sujeita a diversas interferências provenientes do ambiente externo.

3.4.3 Robustez

Ondas de rádio e campos magnéticos provenientes do ambiente externo podem comprometer os dados passados da antena ao leitor RFId. Caso os dados venham corrompidos por estas interferências o sistema deverá ser robusto e permanecer operante sem apresentar nenhum tipo de falha.

3.4.4 Manutenibilidade

O sistema deve permitir a inclusão, edição, listagem e remoção de dados de usuário sem a necessidade de recompilação do módulo de identificação humana para facilitar a manutenção do sistema.

O sistema também deve permitir a inclusão de módulos para o recebimento de informações sobre eventos de usuário (presença ou ausência física nas proximidades do terminal de acesso) sem necessidade de recompilação do módulo de identificação.

3.4.5 Interoperabilidade

O sistema de identificação deve assegurar uma arquitetura flexível a outros equipamentos RFId, não apenas ao equipamento disponível no protótipo.

O sistema domótico deve assegurar conformidade com o protocolo *OpenWebNet* a fim de garantir o controle de qualquer equipamento conectado ao barramento domótico, como a câmera de vídeo, cortinas e luminárias.

3.5 ARQUITETURA DO SISTEMA

3.5.1 Sistema de identificação

Nesta seção iremos mostrar a arquitetura do módulo de identificação, descrevendo as classes que o compõe. Depois explicaremos os principais cenários da aplicação, com ajuda de diagramas construídos com as ferramentas de UML2. No fim há um aprofundamento em alguns aspectos que consideramos mais relevantes. A seguir (Figura 5) tem o esquema de instalação e como o sistema funciona:

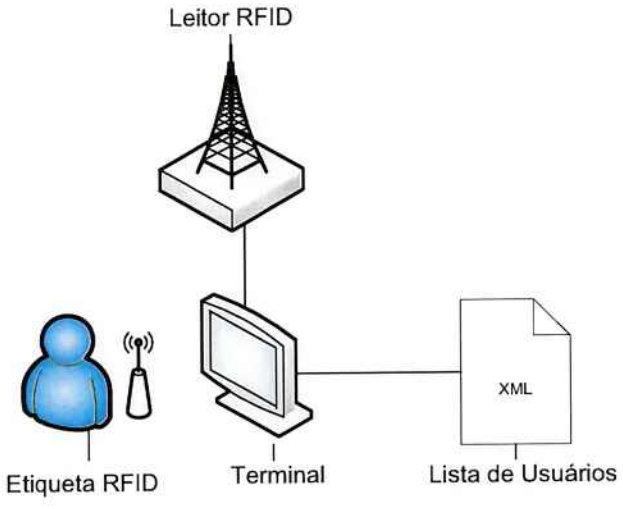

Figura 5 - Esquema de Instalação

A antena RFId instalada próxima ao terminal detecta quando um usuário se aproxima do terminal. O terminal identifica o usuário confrontando o identificador (tag

RFId) recebido pela antena com uma lista de identificadores relacionados com cada usuário.

3.5.1.1 Estrutura Geral

O diagrama de instalação (deployment diagram – Figura 6) é focado em ilustrar apenas o modulo de identificação de usuário. Ele mostra como os componentes estão fisicamente conectados e as dependências entre as bibliotecas do sistema.

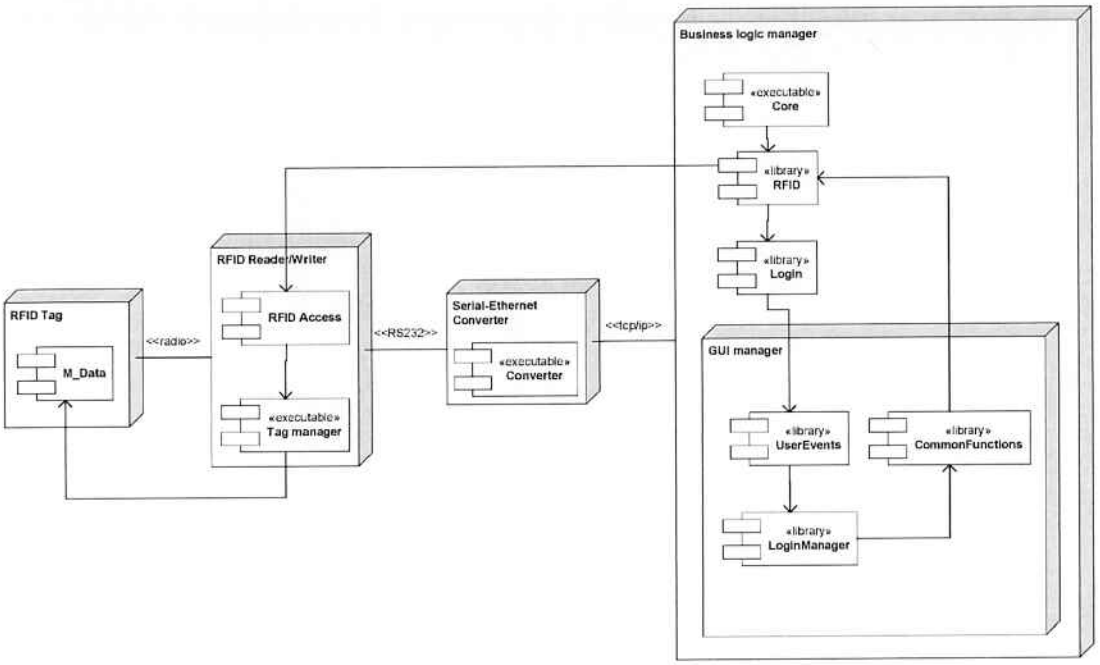

Figura 6 - Visão Estrutural do Sistema de Identificação

O sistema é conectado ao leitor de rádio freqüência através de um adaptador *Serial-Ethernet*. As principais partes do módulo de identificação estão dentro do *Business Logic Manager*. A biblioteca RFId é responsável de comunicar-se com o leitor de rádio freqüência e de chamar funções da biblioteca Login quando necessário. A biblioteca Login é responsável por criar e destruir as sessões de usuário e da comunicação com o *GUI Manager* (Graphic User Interface Manager).

O *GUI Manager* comunica-se com o *Business Logic Manager* de três maneiras possíveis:

1. Requisitando ao equipamento de iniciar ou parar o sistema de varredura (polling) de acesso;

2. Recebendo uma mensagem quando um novo usuário é detectado no raio da antena de rádio freqüência ou quando o mesmo não pode mais ser detectado por um longo período de tempo.

3. Enviando uma requisição para verificar se um dado usuário está presente no raio de leitura da antena. Esta funcionalidade pode ser usada a qualquer momento dentro da aplicação *Flash*.

A biblioteca *CommonFunctions* contém um conjunto de funções que podem ser usadas por qualquer componente do *GUI Manager*. Dentre essas funções nós podemos citar as funções que iniciam ou suspendem a detecção de usuários e a função de verificar a presença de um dado usuário.

A biblioteca *UserEvents* é responsável por receber mensagens sobre a detecção do usuário e então difundir essa mensagem aos outros módulos através do evento *IncomingUser*, no momento em que o usuário entra no sistema, ou do evento *OutcomingUser*, no momento em que o usuário sai do sistema. Os módulos dentro do *GUI Manager* devem se registrar como receptores destes eventos para que a mensagem chegue até eles, podendo assim cada módulo fazer as tarefas necessárias para cada evento. A biblioteca *LoginManager* é um exemplo de módulo receptor. Ele é notificado sobre tais eventos e então decide quando registrar a entrada do usuário no sistema (*login*) ou sua saída (*logout*).

3.5.2 Sistema Domótico

Nesta seção iremos mostrar a arquitetura geral do módulo de automação residencial. Também vamos mostrar os elementos que compõem a maleta da *Bticino* que possui em seu interior o esquema elétrico de uma residencial e como eles estão conectados.

3.5.2.1 Estrutura Geral

O usuário interage com o sistema através do GUI Manager[5], que gera as mensagens que devem ser enviadas para os dispositivos e interpreta as mensagens provenientes dos dispositivos. As mensagens devem passar pelo *OpenWebNet Gateway*, que controla o acesso ao barramento *domótico* (*OpenWebNet BUS*). Ilustramos a seguir (Figura 7) como o usuário interage com o sistema:

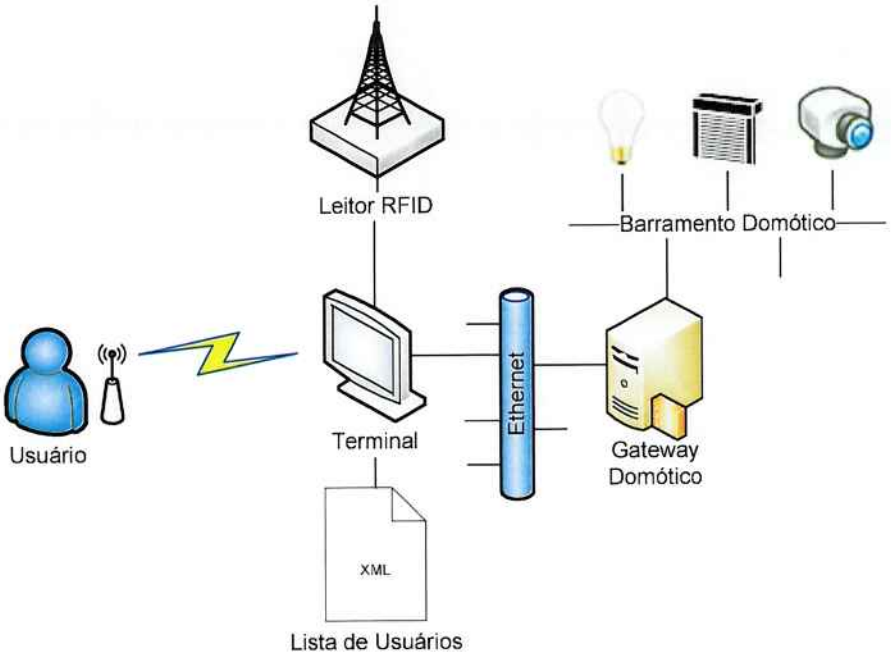

Figura 7 - Estrutura Geral do Sistema domótico

A o diagrama a seguir (Figura 8) apresenta a visão estrutural e está representando apenas o módulo que estamos descrevendo. Ele mostra como os componentes são fisicamente conectados e quais as dependências entre as bibliotecas dentro do sistema.

---

[5] *GUI Manager: Graphical User Interface* – Interface Gráfica para o Usuário

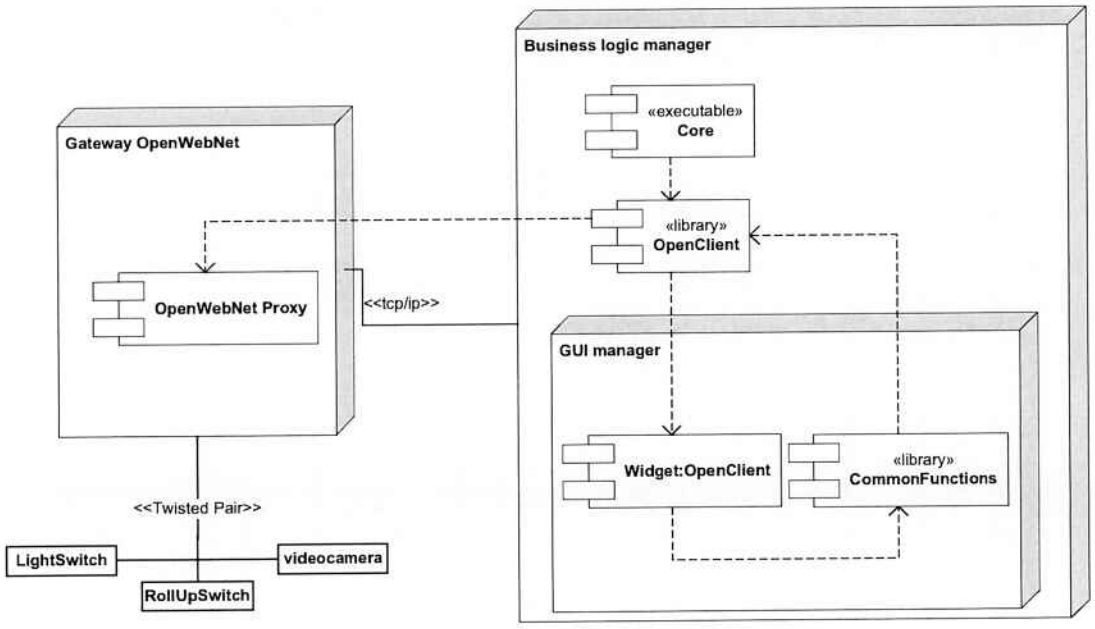

Figura 8 – Visão Estrutural do Sistema de Automação Residencial

O sistema está ligado ao *OpenWebNet Gateway* através de uma conexão *TCP/IP*, que poderia ser feita através de uma rede local ou remotamente, como por exemplo através da Internet. A biblioteca *OpenClient* é responsável por comunicar-se com o *OpenWebNet Gateway*, gerindo as sessões *TCP/IP*. O *OpenWebNet Gateway* contém um servidor web embutido, construído com base na plataforma Linux. Ele está diretamente ligado ao *OpenWebNet BUS*, um par trançado de comunicação, o que permite sua interação com todos os dispositivos conectados.

O *GUI Manager* (Aplicação Flash) comunica-se com os dispositivos de duas formas:

1. Enviando comandos de um determinado dispositivo como, por exemplo, ligando uma luz;

2. Enviando requisições (consultas) para saber o atual estado de um determinado dispositivo ou um grupo de dispositivos.

A biblioteca *CommonFunctions* possui um conjunto de funções que podem ser usadas por qualquer componente do *GUI Manager*. Entre essas funções, podemos destacar as funções de enviar um comando ou para enviar uma consulta.

4 METODOLOGIA

Neste capítulo será analisado o processo de desenvolvimento do projeto, quais as fases envolvidas, qual foi o planejamento feito para atingir os objetivos de cada fase e finalmente como se deu a metodologia de desenvolvimento do sistema de identificação e do sistema de automação residencial.

4.1 METODOLOGIA DE PROJETO

O primeiro passo realizado neste projeto foi identificar as necessidades que guiariam a solução a ser desenvolvida. Esta etapa foi obtida em conjunto com outros integrantes do projeto *Autonomamente*. Após a identificação das necessidades, ou seja, o detalhamento do problema, partimos para o levantamento de requisitos do sistema que solucionaria o problema. A partir desses requisitos as funcionalidades e características do sistema foram obtidas, guiando todo o trabalho que se seguiu.

Tendo essa primeira visão do projeto em mãos, foram realizadas pesquisas de tecnologias e conceitos, a fim de procurar meios para a realização da solução. Esta fase foi de extrema importância, uma vez que muitas das tecnologias do projeto eram pouco conhecidas pelos envolvidos. Assim, nesse momento, um documento bem detalhado da arquitetura proposta do projeto foi elaborado.

Para comprovar e para enriquecer a arquitetura inicialmente definida, um protótipo inicial foi desenvolvido. Em seguida foram realizadas modificações na arquitetura principal com o intuito de resolver e refinar a arquitetura em frente aos problemas encontrados no funcionamento do protótipo. Também foram adicionados documentos detalhando os cenários de utilização dos sistemas de modo a ilustrar como seria o funcionamento ideal do sistema frente a situação principal para a qual os sistemas foram desenvolvidos. Tudo isso foi feito para satisfazer, da melhor maneira possível, as necessidades estipuladas no início do projeto.

Com a visão mais clara do projeto, foi possível elaborar um plano que englobava todo o sistema de forma a realizar etapas em paralelo ou em seqüência dependendo da relação entre elas. Assim tivemos as seguintes fases:

1) Identificação de características e funções do sistema.
2) Pesquisa acadêmica de tecnologias que poderiam ser empregadas para atender as necessidades do sistema.
3) Especificação da arquitetura geral do sistema e seus componentes.
4) Realização do protótipo inicial da arquitetura proposta.
5) Re-elaboração da Arquitetura.
6) Definição dos requisitos do aplicativo do sistema.
7) Codificação do aplicativo.
8) Testes e avaliação.

Através de uma análise mais profunda das necessidades dos usuários e dos outros módulos que se comunicam com nosso sistema, foram então definidos os requisitos não funcionais.

Para iniciar a codificação do aplicativo houve a necessidade de definir a arquitetura final do *software,* para que os módulos dos outros integrantes do projeto *Autonomamente* interagissem de maneira correta com nosso sistema. Desta forma, foram definidas as classes, suas funções e interfaces com os demais módulos do projeto Autonomamente. Esta fase foi de extrema importância para que houvesse um trabalho simultâneo de todos os integrantes do projeto e em conjunto na codificação do aplicativo.

Com essa arquitetura à disposição e o apoio de uma ferramenta de controle de versões, foi possível codificar o aplicativo. Este foi composto especificamente por classes de funcionalidades e classes de interface com o usuário, compondo pacotes como definido na arquitetura do aplicativo.

Por fim, os testes do sistema como um todo foram realizados a fim de identificar problemas e corrigi-los. Estes testes foram realizados com base nos requisitos funcionais e não funcionais e com base nos cenários de utilização principais previamente definidos na arquitetura geral do sistema. Logo, todos os cenários pensados previamente para utilização do sistema foram testados, além de novos cenários, especialmente aqueles referentes a exceções do sistema.

Já a monografia é composta da análise desses documentos obtidos, de referências pesquisadas e indicadas e de reuniões e definições de projeto obtidas em grupo. Este documento final começou a ser elaborada em paralelo com o documento de arquitetura do aplicativo e só foi encerrado ao final dos testes do sistema.

Quanto à implementação do sistema de identificação e de automação residencial foi necessário, durante o processo, a disponibilidade dos equipamentos RFId (emprestados pela *Politecnico di Milano*) e dos equipamentos de automação residencial (doados pela *BTicino* na forma de uma maleta que possui a infra-estrutura básica de uma residência automatizada).

Como tais equipamentos não puderam ser trazidos para o Brasil, desenvolvemos simuladores para os equipamentos de RFId e de automação residencial a fim de poder mostrar de forma mais interativa o funcionamento de nosso sistema.

5 PROJETO E IMPLEMENTAÇÃO

Neste capitulo, serão abordados tópicos referentes ao projeto do sistema de identificação por RFId para uma aplicação de automação residencial, e questões envolvendo a implementação da prova de conceito do mesmo.

5.1 SISTEMA DE IDENTIFICAÇÃO HUMANA POR RFID

Esta seção descreve as classes projetadas para controlar o dispositivo de RFId, bem como criar o *framework* com o qual a interface gráfica (Adobe Flash) irá interagir.

Como citado anteriormente, esta solução faz uso de uma pulseira que identifica o usuário que está interagindo com o sistema. O sistema de varredura (polling) de identificação foi desenhado para identificar os rórulos RFId pertencentes a cada usuário e fazer o gerenciamento da sessão do usuário. Nesta seção, descreveremos como uma sessão de usuário funciona, quando a mesma é criada e depois deixa de existir.

5.1.1 Descrição das Classes

Para lidar com o sistema de varredura (polling), as seguintes classes foram desenvolvidas. Por simplicidade, métodos e atributos ficaram ocultos neste Diagrama de Classe (Figura 9).

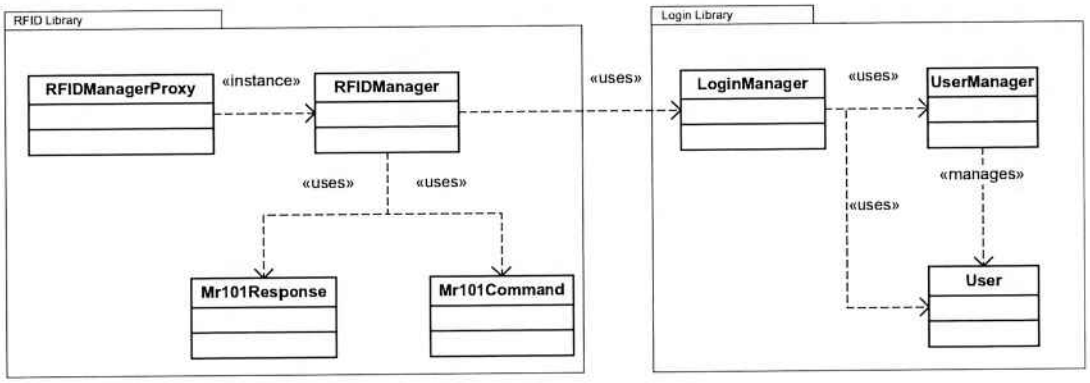

Figura 9 - Diagrama de Classe das Bibliotecas do Sistema de Identificação

A seguir apresentamos uma explicação detalhada sobre as classes consideradas:

Classe RFIDManagerProxy

Esta classe (Figura 10) é responsável por receber as chamadas vindas da aplicação gráfica (Adobe Flash) através dos métodos StartLitening(), StopListening() e VerifyUserPresence(). Esta classe foi criada para garantir que a interface gráfica criará apenas uma instância da classe RFIDManager, evitando a criação de mais de uma varredura (polling) de acesso.

LoginRFID::**RFIDManagerProxy**
-rfidManager : RFIDManager = RFIDManager.Instance
+RFIDManagerProxy() +StartListening() +StopListening() +VerifyUserPresence(in queryRF : string) : ArrayList

Figura 10 - Classe RFIDManagerProxy

Os principais métodos desta classe são:

Método StartListening(): permite ao *Flash* habilitar o leitor de rádio freqüência e iniciar o sistema de varredura (polling) de acesso.

Método StopListening(): desabilita o leitor de rádio freqüência e libera os recursos usados no varredura (polling) de acesso.

Método VerifyUserPresence(): retorna a aplicação gráfica se um usuário está presente no sistema.

Classe RFIDManager

A classe RFIDManager (Figura 11) implementa o sistema de varredura (*polling*) de acesso conectando com o leitor e, periodicamente, procurando pelos rótulos RFId do usuário e detectando a presença de novos rótulos.

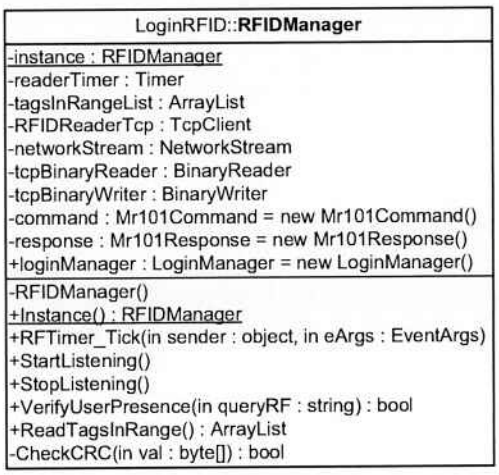

Figura 11 - Classe RFIDManager

Os principais métodos desta classe são:

Método StartListening(): inicia o leitor de rádio freqüência, deixando-o pronto para leitura. Também inicia o cronômetro que, periodicamente, chama o método RFTimer_Tick, o qual lê todas etiquetas RFIDs na região do leitor e chama o LoginManager quando um novo identificador é encontrado.

Método StopListening(): suspende o cronômetro e libera os recursos usados na conexão com o leitor.

Método ReadTagsInRange(): comunica-se com o leitor de RFId, através do uso de um série de mensagens. As mensagens enviadas pela aplicação ao leitor são criadas através da classe Mr101Command, e todas as respostas do leitor são verificadas quanto a sua formação (método CheckCRC()) e as formadas corretamente são processadas utilizando-se a classe Mr101Response.

Classe Mr101Command e Classe Mr101Response

Mr101Command e Mr101Response são utilizadas para, respectivamente, gerar e consumir as mensagens usadas na comunicação com o leitor RF (Figura 12).

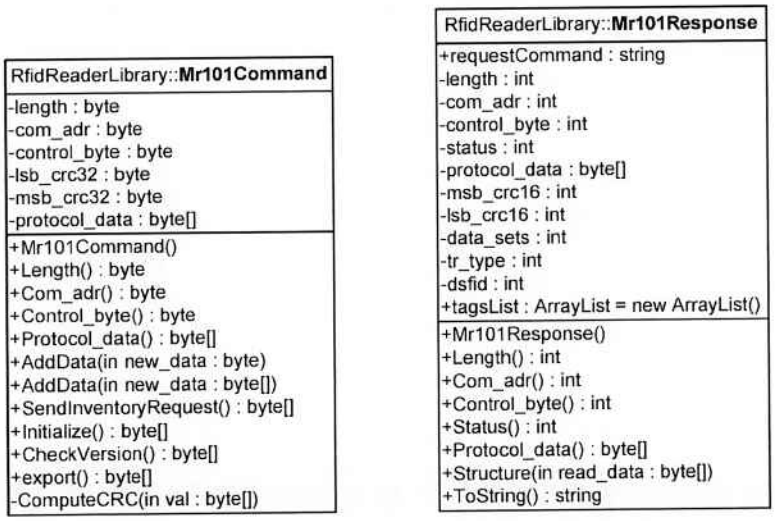

Figura 12 - Classes Mr101Command e Mr101Response

Os principais métodos destas classes são:

Método Export(): gera um série de bytes correspondente ao comando que será enviado, usando o protocolo padrão adotado pelo leitor de RF. O conteúdo da mensagem é criado através dos métodos Initialize(), CheckVersion e SendInventoryRequest().

Método Structure(): recebe um *array* de bytes e o consome, criando um pacote usado para reconhecer as etiquetas RFIds presentes.

Classe LoginManager

A classe LoginManager (Figura 13) gerencia os usuários do sistema, registrando no sistema a sua entrada, quando seu identificador é detectado, e a sua saída depois de um intervalo determinado de tempo correspondente ao tempo de sessão do usuário.

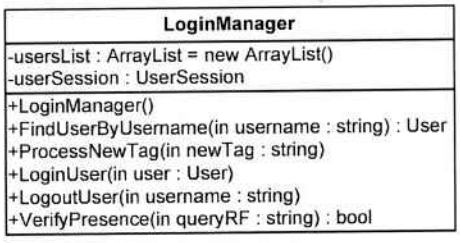

Figura 13 - Classe LoginManager

O conceito de sessão foi usado devido ao fato que algumas vezes o leitor não detectar corretamente todos os identificadores RFId na área de leitura. Este modelo

considera então o usuário presente mesmo se seus identificadores não forem detectados em um dado instante de tempo.

Os principais métodos dessas classes são:

Método ProcessNewTag(): recebe o ID da etiqueta RFId, localiza o usuário correspondente e cria a sessão deste usuário, caso ainda não tenha sido criada (através do método LoginUser). Caso a sessão já exista, o sistema irá apenas atualizá-la.

Método LoginUser(): notifica à aplicação gráfica sobre a entrada do usuário no sistema e cria sua sessão.

Método LogoutUser(): notifica a aplicação gráfica sobre a saída do usuário do sistema, destrói sua sessão e libera os recursos utilizados para tal.

Método VerifyPresence(): responde se um dado usuário é o que atualmente possui a sessão.

Classe UserSession

A classe UserSession gerencia as sessões do usuário. Quando a sessão de usuário expira, essa classe notifica a classe LoginManager que irá registrar a saída do usuário do sistema.

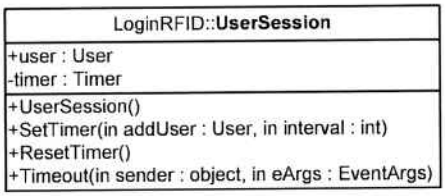

Figura 14 - Classe UserSession

Os principais métodos dessas classes são:

Método ResetTimer(): reinicia a sessão do usuário. Este método é chamado pela classe LoginManager, quando é detectado um identificador RFId pertencente ao usuário que atualmente detém a sessão.

Método Timeout(): notifica a classe LoginManager que a sessão do usuário acabou.

Classe User

A classe *User* (Figura 15*)* guarda as informações sobre o usuário, como seu *username* e o conjunto de possíveis rótulos RFIds que estão atrelados ao usuário.

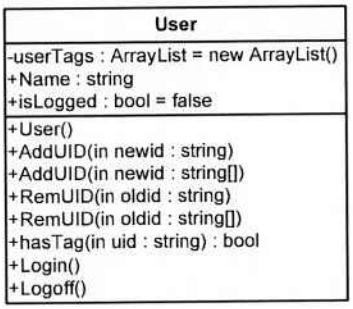

Figura 15 - Classe User

O principal método desta classe é:

Método hasTag(): retorna *true* (verdadeiro) se o usuário é o proprietário de uma certa etiqueta RFId e *false* (falso) caso contrário.

5.1.2 Cenário Principal

O principal cenário para identificação através do RFId compreende três funções disponíveis ao *GUI Manager*: *Start* (iniciar), *verify user presence* (verificar presença de usuário) e *Stop* (parar).

No diagrama de seqüência (Figura 16) ilustramos os módulos envolvidos neste cenário e como eles atuam após a requisição da aplicação Flash. Após a requisição de inicio (*Start*), o algoritmo de varredura (*polling*) entra em execução até o momento em que uma requisição para parar (*Stop*) é recebida.

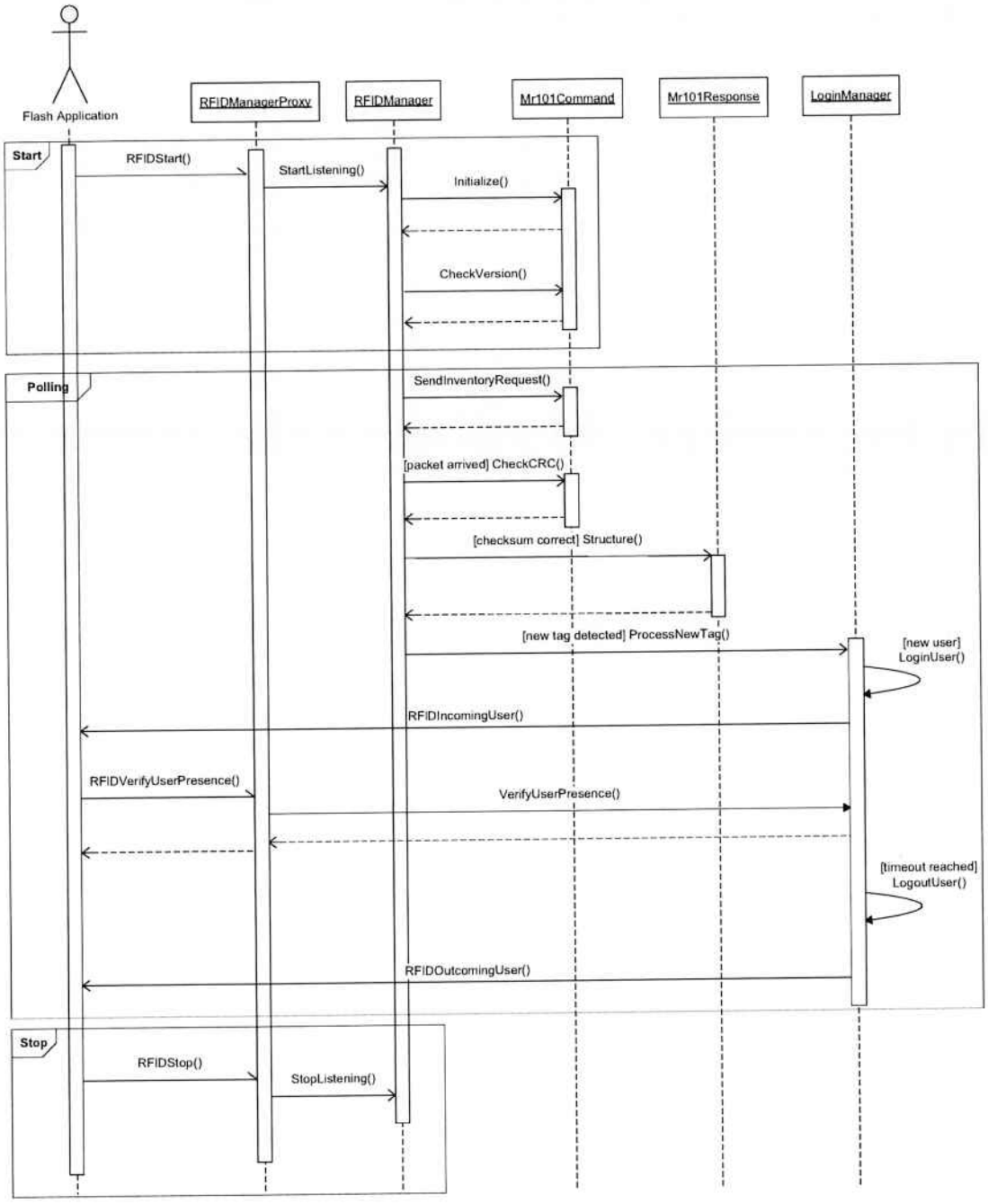

Figura 16 - Diagrama de seqüência do Cenário Principal

O módulo principal é o *RFIDManager*, que realiza a conexão com o leitor de RFId e sua inicialização, coordena a comunicação com este equipamento e executa o sistema de varredura (*polling*) para a detecção de etiquetas (*tags*) RFId.

5.1.3 Sistema de Varredura para Identificação de Acesso

O diagrama lógico abaixo ilustra como o sistema de varredura funciona, explicando exatamente todas as decisões tomadas dependendo de certas condições.

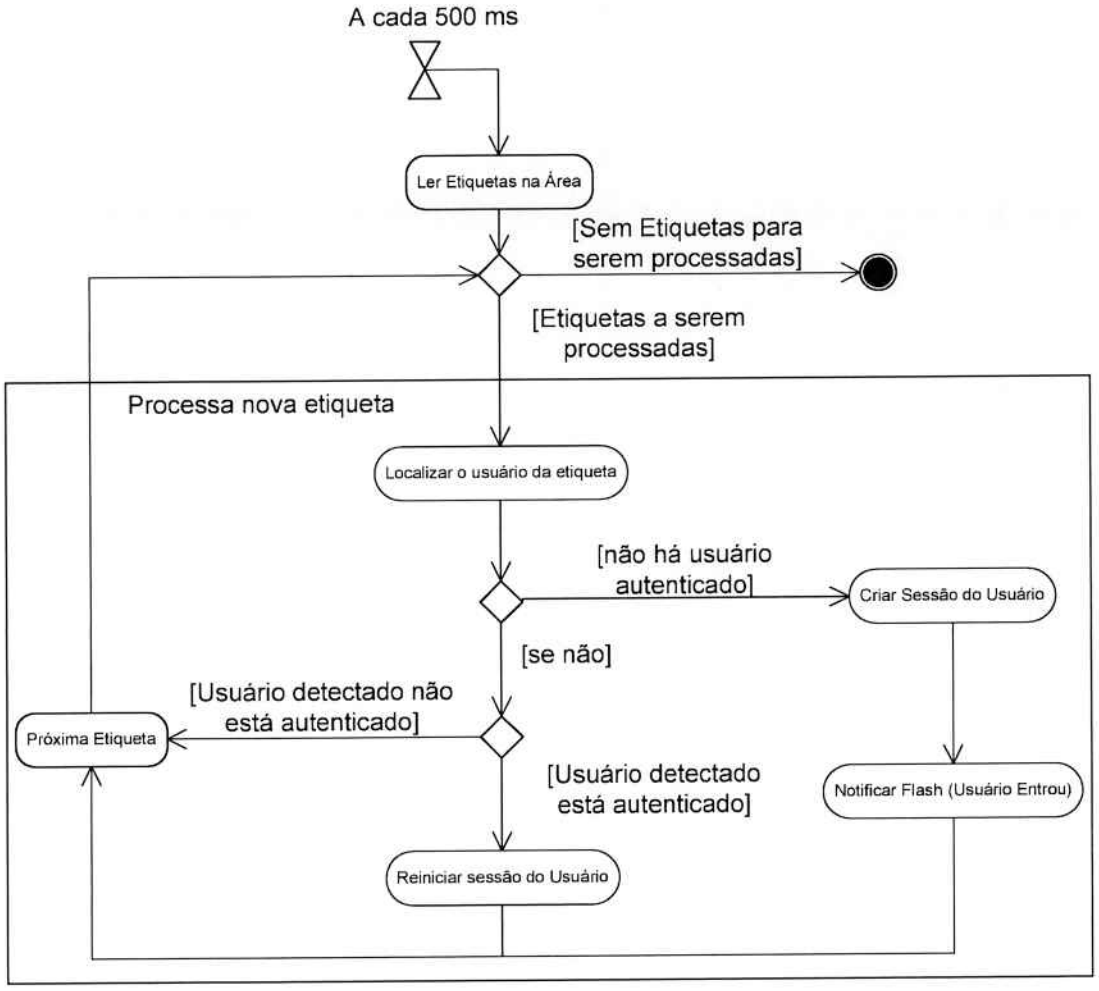

Figura 17 - Diagrama Lógico do Sistema de Varredura para Identificação de Acesso

A varredura inicia-se a cada intervalo de tempo definido no arquivo de configuração. O tempo padrão entre duas varreduras é de quinhentos milissegundos. O primeiro passo da varredura é a leitura de todas as etiquetas RFIds dentro do raio de leitura da antena de RFId. O sistema então localiza o usuário que detêm a etiqueta RFId detectada e, caso não existam usuários logados até o momento, inicializa a sessão de usuário que conterá suas informações. Após a criação da sessão, o sistema avisa à aplicação gráfica sobre o usuário que está

entrando no sistema. Se a etiqueta detectada pertence ao usuário que atualmente detêm a sessão, o sistema reinicializa o cronometro da sessão, caso contrario o sistema apenas segue para o processamento da próxima etiqueta detectada.

O diagrama a seguir (Figura 18) ilustra o ciclo de vida da sessão de usuário, iniciada quando um novo usuário é detectado.

Figura 18 - Ciclo de Vida da Sessão de Usuário

A sessão dura ao menos o tempo definido pelo parâmetro *UserTimeout*. O cronometro que controla a sessão será reiniciado toda vez que uma etiqueta RFId pertencente ao usuário identificado é detectada. O valor padrão do parâmetro *UserTimeout* é de quinze segundos.

5.1.4 Plataforma de desenvolvimento

O sistema foi dividido em duas partes principais, o *GUI Manager* (gerenciador de interface) e o *Business Logic Manager* (gerenciador de lógica de negócio). Cada parte do sistema foi desenvolvida na plataforma que melhor se encaixava com suas necessidades. O *Business Logic Manager* foi desenvolvido na plataforma Microsoft C# .Net enquanto o *GUI Manager* foi desenvolvido usando o Adobe Flash.

5.1.5 Protótipo

Para testar o módulo foi criado o protótipo com uma interface que permite ao usuário interagir diretamente com todos os métodos implementados. A Interface Gráfica do Usuário (Figura 19) a seguir ilustra a interface implementada.

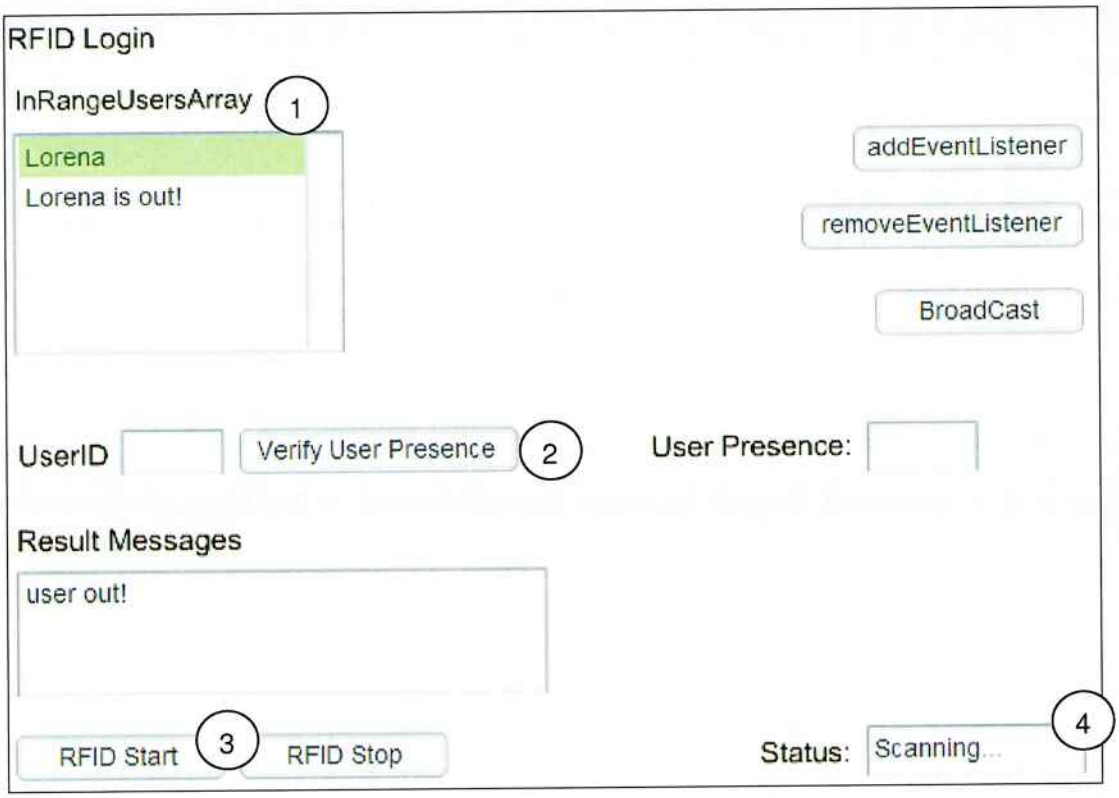

Figura 19 - Interface Gráfica do Usuário para o módulo de RFId

Nós podemos destacar quatro regiões nesta interface:

1. *InRangeUsersArray*: Neste campo nós podemos observar os usuários que entraram ou saíram da área de alcance da antena RFId;
2. *Verify User Presence*: Permite perguntar ao *framework* se um dado usuário está presente na área da antena RFId;
3. *RFId Start* e *RFId Stop*: Permite acessar as funções do *framework* que iniciam e interrompem o sistema de varredura de acesso.
4. *Status*: Exibe o *status* do leitor RFId.

5.1.6 Aplicação final utilizando o *Framework*

Após os testes, estava tudo pronto para usar o *framework* na aplicação final. A figura abaixo (Figura 20) mostra como o *framework* foi utilizado dentro do sistema.

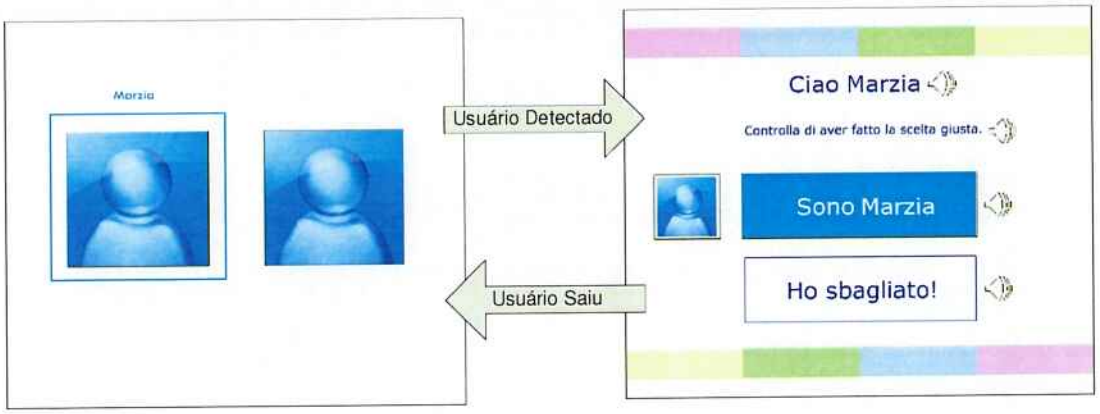

Figura 20 - Tela de Autenticação e Tela Inicial

O *framework* informa a interface gráfica quando um usuário se aproxima e quando este não está mais na área de alcance da antena por um longo período de tempo. O *framework* também permite ao *GUI Manager* a perguntar se um dado usuário tem sua sessão ativa e também controlar o leitor RFId, iniciando ou interrompendo o sistema de varredura.

5.1.7 Restrições de Projeto e Implementação

Algumas restrições externas foram respeitadas para o correto comportamento do sistema de identificação inserido no projeto *Autonomamente*:

Restrição	Descrição
Distância de Leitura	O leitor RFId deve ser apto a ler uma etiqueta RFId a uma distância máxima de 60 cm em relação ao centro da antena, e a uma distância mínima de 1 cm.
Freqüência de Leitura	O leitor RFId deve ser apto a ler todas as etiquetas RFId na área de alcance a uma freqüência mínima de 2 vezes por segundo (a cada intervalo de 500ms).
Atraso na Identificação do Usuário	O leitor RFId deve ser apto a identificar o usuário em menos de 2 segundos.
Atraso na Inicialização	O leitor RFId deve ser apto a iniciar a leitura das etiquetas RFIds em um intervalo máximo de 6 segundos.

Tabela 4 - Restrições

5.1.8 Parâmetros de Configuração

Para facilitar a adaptação do sistema ao ambiente o qual o mesmo será instalado, alguns parâmetros podem ser modificados:

Parâmetro		Descrição
Leitor RFId	ReaderIP	Endereço IP do leitor.
	ReaderPort	Porta do leitor. O valor padrão é 7001.
	ScanInterval	Intervalo de tempo entre 2 leituras, em ms.
Usuário	UserTimeout	Tempo o qual o usuário permanece registrado (duração da sessão) depois que foi detectado, em ms.

Tabela 5 - Parâmetros de Configuração

Estes parâmetros são armazenados dentro de um XML (*Extended Markup Language*) reservado para a configuração do módulo de identificação RFId.

5.2 SISTEMA DOMÓTICO

Nesta seção é apresentado a especificação, a análise dos requisitos, a arquitetura do *software* projetado e, finalmente, como foi implementado e testado o *software*.

5.2.1 Descrição das Classes

Para lidar com o barramento domótico, as seguintes classes foram desenvolvidas.

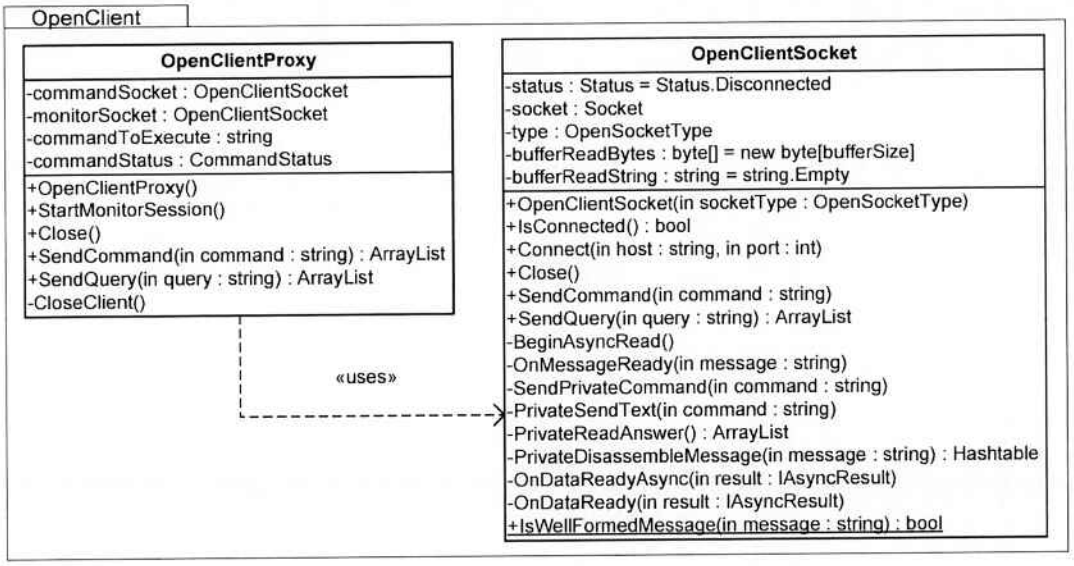

Figura 21 - Pacote Open Client

Uma explicação detalhada de cada classe é apresentada a seguir:

Classe OpenClientProxy

Esta classe é responsável por receber as chamadas da aplicação Flash através dos métodos *StartMonitorSession* (), *Close* (), *SendCommand* () e *SendQuery* (), que são os principais métodos desta classe. As conexões com o *OpenWebNet Gateway* são feitas indiretamente através da classe *OpenClientSocket*.

Método *StartMonitorSession*(): conecta com o *OpenWebClient Gateway* e inicializa a sessão TCP/IP no modo de monitoramento. Neste modo, o cliente recebe mensagens do portal sobre todas as ações efetuadas no *Automation BUS*;

Método *Close*(): fecha uma conexão inicializada pelo método *StartMonitorSession*;

Método *SendCommand* (): cria uma nova conexão com o *OpenWebNet Gateway* no modo de comando, envia um comando e retorna se o mesmo foi executado ou não.

Método *SendQuery*(): cria uma nova conexão com o *OpenWebNet Gateway* no modo comando, envia uma consulta e retorna um objeto que representa a matriz respondida pelo *OpenWebNet Gateway*. Geralmente essa matriz indica o estado dos dispositivos apontados pela consulta.

Classe OpenClientSocket

Esta classe é responsável por conectar-se com o *OpenWebNet Gateway*. Possui um conjunto de métodos utilizados para lidar com a troca de mensagens, respeitando o protocolo *OpenWebNet*.

Os principais métodos desta classe são:

Método *Connect* (): através do método *OpenClientSocket*, cria uma conexão com o *OpenWebNet Gateway*, endereçado pelo par de argumentos: endereço e porta de conexão. Este método é chamado pelo *OpenClientProxy* ao iniciar uma conexão no modo de monitoração ou no modo de comando.

Método *OpenClientSocket* (): método utilizado pelo método Connect para criar a conexão com o *OpenWebNet Gateway*.

Método *SendCommand* (): cuida da comunicação necessária para enviar um comando ao *OpenWebNet Gateway*. Responde se o comando solicitado foi enviado e executado com sucesso.

Método *SendQuery* (): chamado quando uma consulta deve ser enviada ao *OpenWebNet Gateway*. Ele envia a consulta e trata as respostas provenientes do *OpenWebNet Gateway*, criando um objeto que contém a lista dos dispositivos e seus status atuais.

Método *IsWellFormedMessage* (): executa uma verificação se uma mensagem enviada ou recebida está de acordo com o protocolo *OpenWebNet*. É utilizado pelos métodos *SendCommand* () e *SendQuery* ().

Método *PrivateDisassembleMessage* (): trata a mensagem recebida do *OpenWebNet Gateway* e a desmonta, identificando o ID do dispositivo e a sua situação atual.

5.2.2 Cenário Principal

O cenário principal para automação residencial que fará do uso deste módulo prevê a utilização de duas funções disponibilizadas a aplicação *Flash*: enviar um comando (*send command*) e enviar uma consulta (*send query*). O diagrama de seqüência (Figura 22) apresenta como os diversos módulos estão envolvidos neste cenário e como eles atuam após uma requisição feita pela *GUI Manager* (Flash).

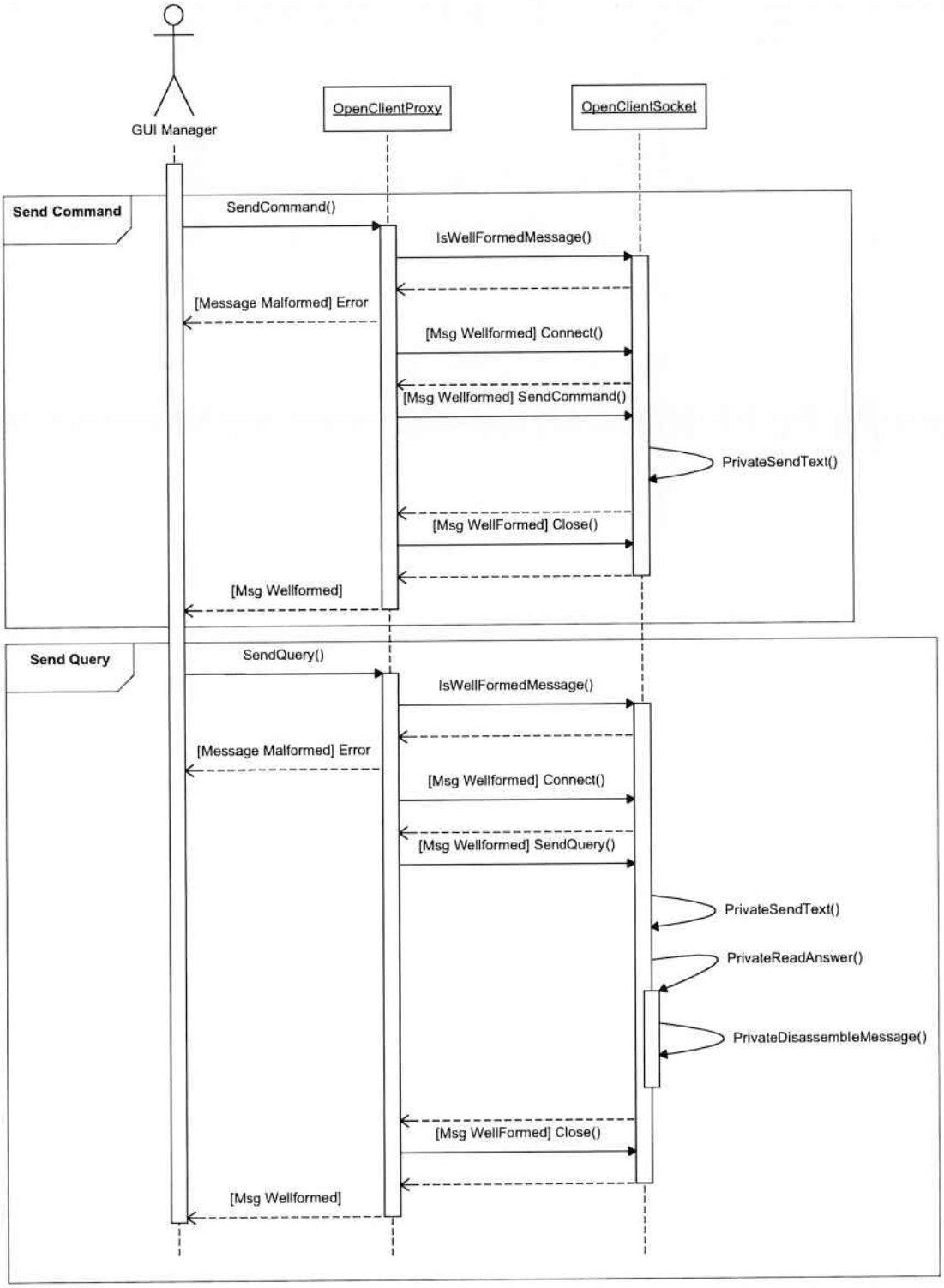

Figura 22 - Cenário Principal

O modulo principal chama-se *OpenClientSocket*. Ele é responsável pó conectar-se com o *OpenWebNet Gateway* e de gerir a comunicação com este dispositivo.

A aplicação Flash tem acesso ao *OpenWebNet Gateway* através da classe *OpenClientProxy*. Para o envio de um comando, dentro da classe *OpenClientProxy* existe o método *SendCommand*(). Antes do envio do comando, a classe *OpenClientProxy* verifica se a mensagem está de acordo com o padrão *OpenWebNet*, reportando o erro a quem o chamou no caso em que a mensagem não esteja de acordo com o padrão. As mensagens bem formadas são enviadas através da classe *OpenClientSocket* e a resposta se o comando foi executado ou não é então encaminhado a aplicação *Flash*.

Para a execução de uma consulta, a aplicação Flash utiliza-se do método *SendQuery*(), dentro da classe *OpenClientProxy*. A principal diferença deste método com o anterior é que as respostas bem formadas proveniente do *OpenWebNet Gateway* são enviadas ao método *PrivateDisassembleMessage*(), responsável por criar um vetor contendo uma lista de endereço de dispositivos e o seus respectivos estados. Este vetor é então entregue ao *GUI Manager* (*Flash*).

5.2.3 Plataforma de desenvolvimento

Como citado anteriormente para o módulo de identificação de usuário, o sistema foi dividido em duas partes: o *GUI Manager* (gerenciador de interface) e o *Business Logic Manager* (gerenciador de lógica de negócio). Para o desenvolvimento da interface gráfica utilizou-se o aplicativo Adobe Flash e a lógica de negócio foi desenvolvida dentro da plataforma Microsoft C# .Net.

5.2.4 Protótipo

Para testar o módulo foi criado o protótipo com uma interface que permite ao usuário de diretamente interagir com todos os métodos implementados. Através dela, o usuário pode acender ou apagar uma lâmpada, controlar a cortina e ligar ou

desligar uma câmera de vídeo. A interface a seguir (Figura 23) ilustra a interface implementada.

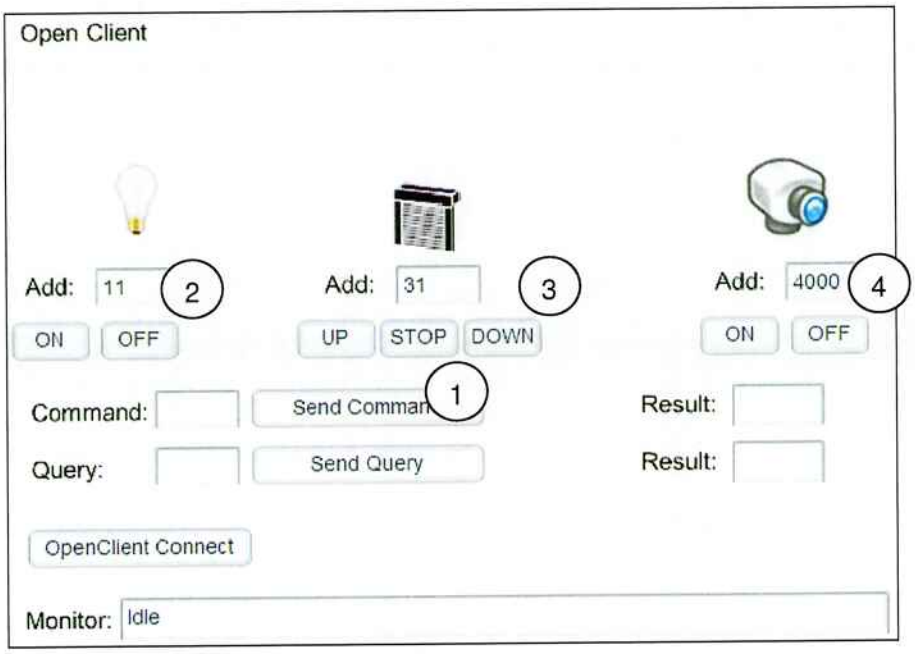

Figura 23 - Interface de Testes para o Sistema Domótico

Nós podemos destacar quatro regiões desta interface:

1. Botões *Send Command* e *Send Query*: estes botões permitem ao usuário enviar um comando ou uma consulta para o *OpenWebNet Gateway*. Para usar esses botões o usuário deve conhecer o protocolo *OpenWebNet*;
2. Botões *ON* e *OFF*: controla o estado de uma lâmpada endereçada pelo campo *Add*.
3. Botões *UP*, *STOP* e *DOWN*: controlam uma cortina endereçada pelo campo *Add*, fazendo-a subir, parar ou descer, respectivamente.
4. Botões *ON* e *OFF*: controlam a vídeo-camera instalada na maleta.

5.2.5 Restrição de Projeto e Implementação

Algumas restrições devem ser respeitadas para o correto comportamento do sistema em termos de tempo de resposta:

Restrição	Descrição
Tempo para enviar um comando	O framework deverá ser capaz de enviar um comando ao gateway em menos de 300 ms.
Tempo para enviar uma consulta	O framework deverá ser capaz de enviar uma consulta e processar a resposta em menos de 2 segundos.
Atraso na conexão	O framework deverá ser capaz de estabelecer a conexão com o gateway em menos de 2 segundos.

Tabela 6 - Restrições

5.2.6 Parâmetros de Configuração

Para facilitar a adaptação do sistema ao ambiente ao qual o mesmo será instalado, alguns parâmetros podem ser modificados.

Parameter	Description
WebserverIP	Endereço IP do OpenWebNet Gateway.
WebserverPort	Porta do protocolo IP do OpenWebNet Gateway. O valor padrão é 20000.

Tabela 7 - Parâmetros de Configuração

6 TESTES E AVALIAÇÃO

Neste capítulo é descrito o ambiente que o protótipo foi testado na Itália e as características e critérios para estes testes, explicando o raciocínio por trás de cada um deles. No final deste capítulo é apresentada uma análise dos resultados obtidos durante a fase experimental, fornecendo um breve comentário sobre eles.

6.1 AMBIENTE DE TESTE

O equipamento utilizado para efetuar o teste foi um computador com processador Intel Celeron M de 1.1Ghz com 512 Mb de RAM, com uma tela LCD sensível ao toque de 15 polegadas. Este equipamento foi montado em um gabinete de metal tipicamente utilizado em ambientes industriais, como mostrado abaixo (Figura 24).

Figura 24 - Foto do protótipo do terminal de usuário

Para testarmos o RFId, foi disponibilizado um leitor de RFIDs da *FEIG Electronic*, modelo ID ISC.MR100 - A, um leitor de médio alcance, que trabalha na

freqüência de 13,56 MHz, com potência máxima de transmissão de 1W. A antena ligada ao leitor é também de *FEIG Electronic*, modelo *ISC.ANT340/320 Pad Antenna*, projetada para aplicações em interiores, com o alcance máximo de leitura de 30 cm. O leitor RFId foi conectado a um conversor Ethernet - Serial, devido ao fato que o leitor dispõe apenas de uma saída RS232 e os terminais geralmente não têm uma porta serial para realizarmos a conexão.

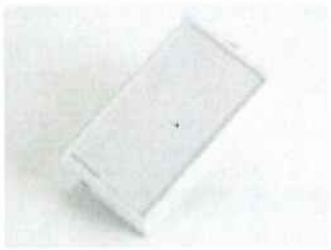

Figura 25 - Leitor RFId

Figura 26 - Tags e Antena

Para testarmos o módulo de automação residencial foi testado com uma mala da *BTicino*, equipada com um servidor e alguns dispositivos, que representam um interruptor de uma fonte luminosa, o controlador de uma cortina automática e uma vídeo-câmera. Todos esses dispositivos estão em conformidade com o protocolo *OpenWebNet*. A imagem a seguir (Figura 27) mostra a maleta e seus componentes.

Figura 27 - Foto da mala contendo o equipamento da Bticino

6.2 BENCHMARKS

É possível dividir os aspectos testados em dois grupos: Identificação do Usuário e Automação Residencial. No grupo de Identificação do Usuário focamos na medição de aspectos relevantes ao comportamento do RFId, como o tempo de resposta e a sua acurácia. Relativamente ao grupo de Automação Residencial os testes foram focados na interação entre o sistema e o bus domótico, como por exemplo, o tempo necessário para se enviar um comando e a porcentagem de comandos executados corretamente.

6.2.1 Identificação do Usuário – (RFId)

Nesta seção é analisado os seguintes aspectos:

- Tempo necessário para inicializar o leitor RFId;
  - Requerido: menos de 6 segundos;
- Numero máximo de identificadores lidos simultaneamente;
  - Requerido: ao menos 6 identificadores;
- Tempo necessário para a leitura de um grupo de identificadores;
  - Requerido: tempo médio menor que 2 segundos;
- Distância máxima de leitura;
  - Requerido: ao menos 20 centímetros;
- Inclinação máxima da etiqueta RFId;
  - Requerido: próximo ao padrão para este tipo de identificador, ou seja, 45° em relação ao plano da antena;
- Acurácia na identificação de um usuário;
  - Requerido: maior que 95%.

Para as medições de intervalo de tempo foram introduzidas algumas mudanças no código do programa, incluindo-se rotinas que armazenam o tempo decorrido para a realização de uma determinada ação. Para se obter a quantidade máxima de etiquetas lidas simultaneamente foi elevada a quantidade de identificadores e observado o número de etiquetas detectadas. O resultado do último

teste foi verificado variando a distância entre a antena e o identificador e observando a leitura realizada pelo sistema.

6.2.1.1 Tempo de Inicialização do leitor RFId

Esta medição tenta estimar o tempo médio que o leitor leva antes de estar pronto para leitura. É o tempo necessário para que o sistema de varredura (*polling*) se inicie. O gráfico abaixo (Figura 28) ilustra os dados coletados e o tempo médio de inicialização.

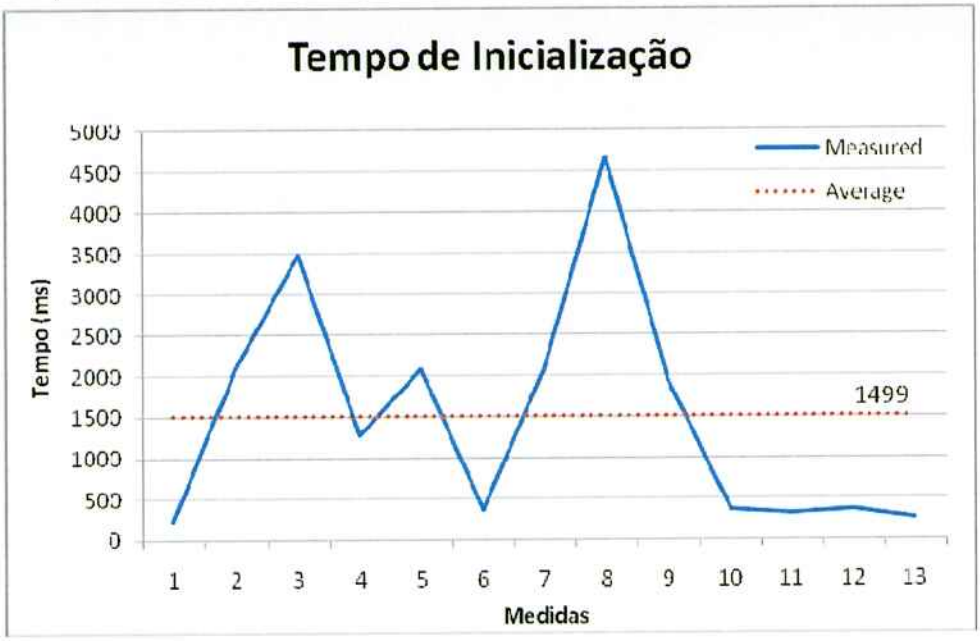

Figura 28 - Tempo de Inicialização

O tempo médio de inicialização do leitor de RFId de acordo com nossas medições foi de 1,5 segundos, menor do que o tempo requerido de 6 segundos. O tempo de inicialização nunca superou os 6 segundos, com um máximo de 4,6 segundos.

6.2.1.2 Número máximo de etiquetas lidas simultaneamente

Esta medição mostra quantas etiquetas RFId o sistema estará apto a identificar simultaneamente. Dado que a pulseira descrita anteriormente tem 4

etiquetas embutidas, foi considerada uma quantidade mínima suficiente para ler duas pulseiras como uma quantidade segura, o que representaria dois usuários na proximidade do terminal.

Durante o experimento foi possível facilmente ultrapassar esse requerimento, estando o sistema apto a ler até 24 etiquetas perfeitamente. Considerando-se 4 etiquetas por pulseira, um total de 6 usuários poderiam ser identificados simultaneamente.

6.2.1.3 Tempo necessário para a leitura de um grupo de etiquetas

Esta medição mostra o tempo necessário para que o leitor identifique todas as etiquetas em seu raio de ação, mostrando como o tempo total cresce com o aumento do número de etiquetas. Para cada quantidade de etiquetas foram realizadas ao menos 15 medições. O gráfico a seguir ilustra os dados coletados:

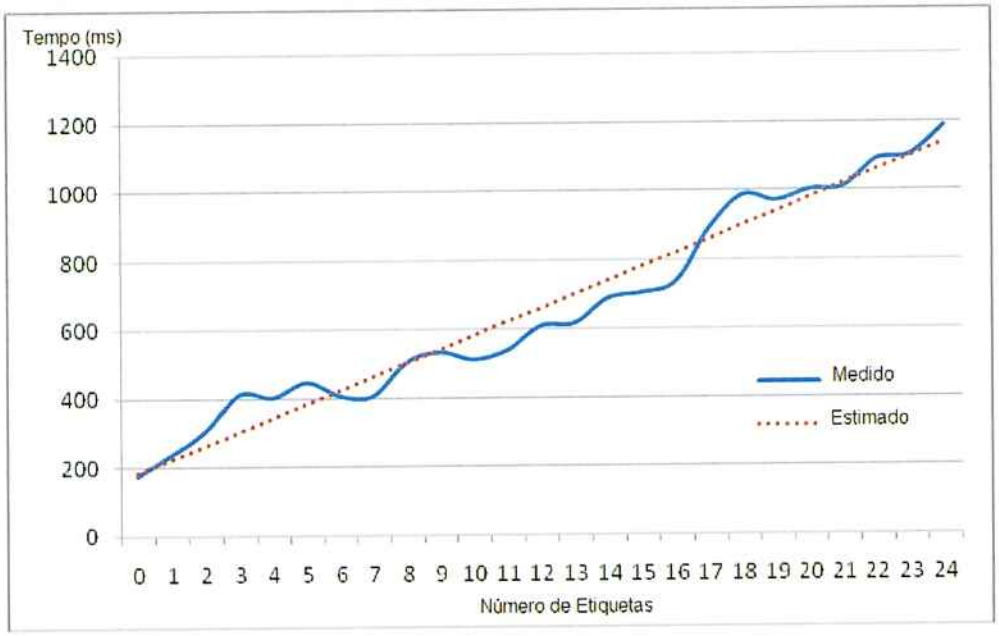

Figura 29 - Tempo para a leitura de um grupo de etiquetas

Foi possível criar uma reta que estima o tempo necessário para ler as informações de um grupo de etiquetas. Esta reta inicia-se com 180ms e cresce 40ms a cada etiqueta RFId adicionada ao grupo. O *framework* é apto a ler todas as etiquetas dentro do intervalo requerido de 2 segundos.

6.2.1.4 Distância máxima de leitura

Um importante aspecto a ser medido é a máxima distância a qual o leitor está apto a identificar uma etiqueta. A idéia inicial do projeto é colocar a antena do leitor RFId bem próxima a tela, detectando a pulseira que o usuário estaria usando no momento em que ele se aproxima do *touch screen*. Durante a sessão de testes, a distância máxima de leitura foi de apenas 14,5 centímetros. O leitor começa a falhar na leitura quando a etiqueta encontra-se a 10,5 centímetros de distância do leitor, aumentando a taxa de erros gradativamente até os 14,5 centímetros. Não foi possível detectar as etiquetas colocadas a uma distância superior aos 15 centímetros. Foi, então, possível observar que a distância máxima de leitura é inferior a requerida (20 centímetros). No final deste capítulo é exposto as possíveis soluções para este problema.

6.2.1.5 Inclinação máxima da etiqueta

Foi realizada a medição da máxima inclinação, em relação ao plano da antena, a qual a etiqueta RFId poderia ser identificada. O esquema a seguir (Figura 30) ilustra melhor o ângulo que foi medido durante nossos testes.

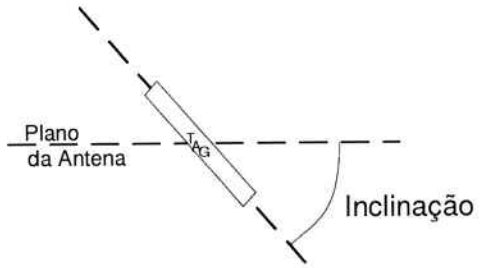

Figura 30 - Inclinação da etiqueta em relação à antena

A máxima inclinação foi medida em quatro diferentes distâncias entre a etiqueta e a antena. O resultado é mostrado na tabela a seguir:

Distância (cm)	Inclinação Máxima
2.5	62°
5	51°
7.5	45°
10	43°

Tabela 8 - Máxima inclinação em relação à distância da antena

Pode-se observar que a máxima inclinação cai conforme a distância em relação ao centro da antena aumenta. O leitor está apto a identificar a etiqueta RFId mesmo quando ela se encontra a uma inclinação próxima aos 45°, o que é um valor padrão para este tipo de antena. O uso da pulseira com etiquetas RFId em diferentes posições ajuda a ultrapassar essa limitação. Outra maneira de se resolver esse problema é através da utilização de leitores com capacidade de geração de ondas circulares polarizadas ou através do uso de etiquetas RFId dotadas de antenas tripolares que, devido a sua geometria, sempre conseguem ao menos uma de suas componentes estar no mesmo plano da antena de leitura [20].

6.2.1.6 Acurácia na identificação de um usuário

Durante a sessão de testes, o *framework* foi capaz de identificar corretamente os usuários em aproximadamente 100% dos casos. Os erros aconteciam apenas quando o usuário mantinha a pulseira por um período de tempo muito curto, geralmente menor do que o intervalo de tempo entre duas leituras.

Uma possível solução para esses casos seria reduzir o tempo entre duas leituras. O intervalo usado nos testes foi de 500ms, mas durante o teste demonstrou-se que um intervalo de 250ms poderia ser usado sem sobrecarregar o sistema.

6.2.2 Automação Residencial

Nesta seção é analisado os seguintes aspectos:

- Tempo necessário para enviar um comando;
    - Requerido: tempo médio menor que 300ms;

- Tempo necessário para enviar uma consulta;
  - Requerido: tempo médio menor que 2 segundos;
- Tempo necessário para inicializar a conexão e enviar o primeiro comando;
  - Requerido: tempo médio menor que 2 segundos.

Todas essas medições foram obtidas através da introdução de algumas alterações no código, incluindo-se rotinas que determinam o tempo gasto para a execução de uma dada tarefa. Para cada requisito foram executadas ao menos 20 medições.

6.2.2.1 Tempo necessário para enviar um comando

Este teste mostra o tempo necessário para o *framework* enviar um comando e sua execução pelo equipamento de automação. É importante saber o quão rápido o sistema responde as solicitações do usuário. Um tempo de resposta curto é importante para dar ao usuário a sensação de que o sistema está respondendo instantaneamente [6].

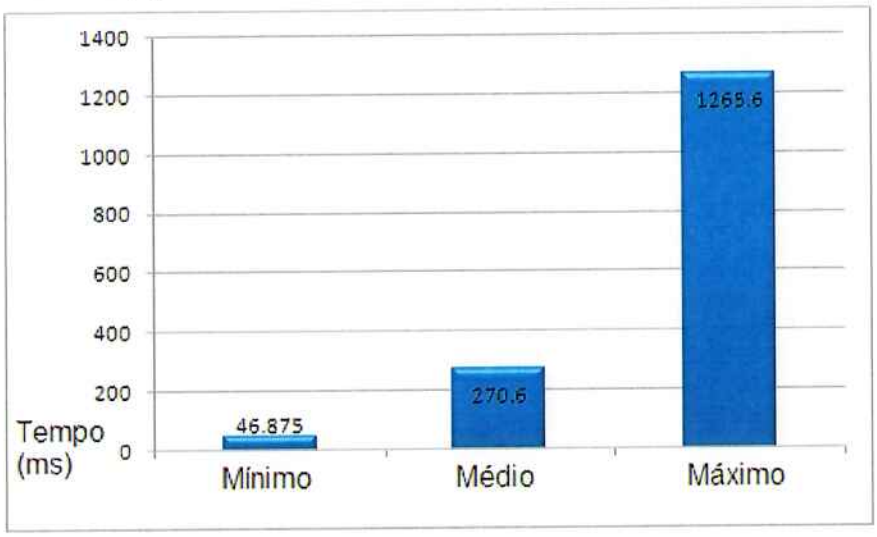

Figura 31 - Tempo mínimo, médio e mínimo para envio de um comando

Os dados coletados nos mostram que o *framework* geralmente responde em menos de 60ms (83% dos casos), mas algumas vezes o sistema levou aproximadamente 1 segundo para responder. Uma importante melhoria futura seria ajustar a conexão para garantir que o sistema responda rapidamente.

6.2.2.2 Tempo necessário para enviar uma consulta

Este teste mostra o tempo necessário para o *framework* enviar uma consulta e receber a resposta do equipamento de automação.

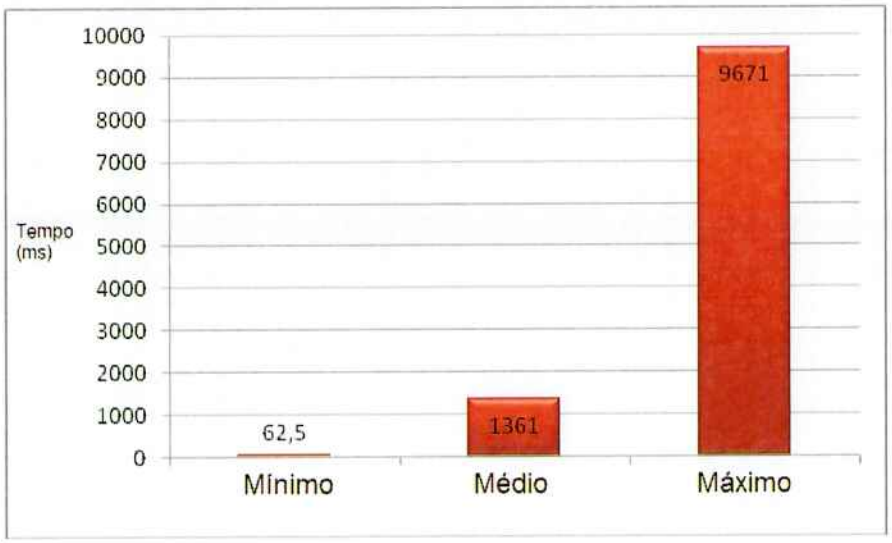

Figura 32 - Tempo mínimo, médio e máximo para envio de uma consulta

Os dados coletados nos mostram que o *framework* geralmente responde em menos de 200ms (75% das vezes), mas algumas vezes ele gasta aproximadamente 2, 3 e em um caso 10 segundos para responder a consulta. Uma importante melhoria futura seria ajustar a conexão para garantir que o sistema responda sempre no tempo adequado.

6.2.2.3 Tempo necessário para inicializar conexão e enviar o primeiro comando

Como antes do envio do primeiro comando o *framework* deve criar a conexão com o servidor da Bticino, é importante se considerar o tempo gasto pelo framework para inicializar a conexão. Infelizmente não era possível medir o tempo de conexão separadamente, mas é possível estimá-lo indiretamente medindo o tempo para envio do primeiro comando, quando a conexão é criada.

O gráfico da Figura 33 ilustra nossas medições:

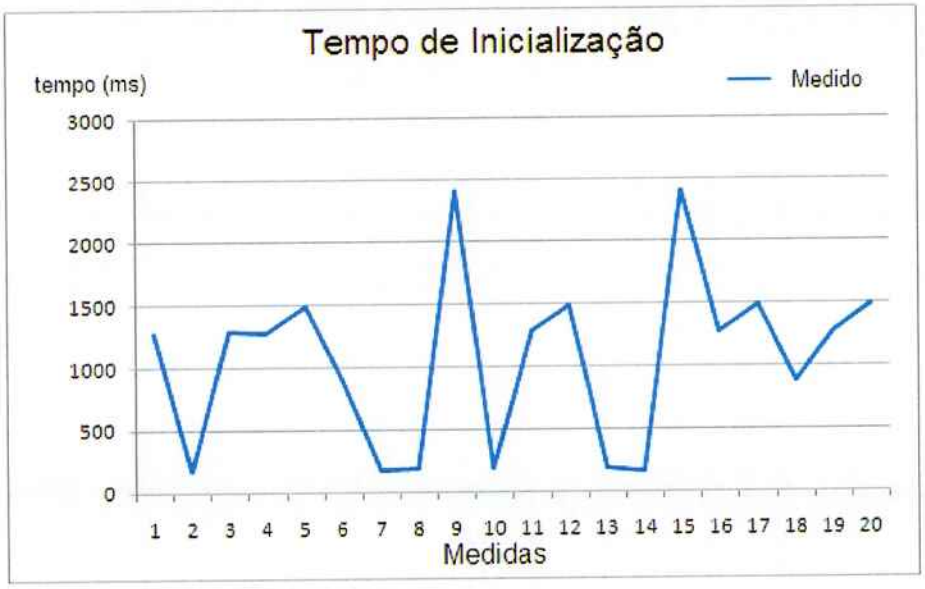

Figura 33 - Inicialização da Conexão

O tempo médio de inicialização foi de aproximadamente 1,06 segundos, abaixo do valor máximo de 2 segundos dos nossos requisitos. Aparentemente em alguns casos o servidor parece estar ocupado e demora mais tempo para aceitar a tentativa de conexão.

6.3 RESULTADOS

A avaliação realizada com o protótipo deu a possibilidade de verificar o comportamento do *framework* em um ambiente próximo ao de sua aplicação final. Foi possível analisar se o *framework* responde de acordo com o que esperávamos. Para melhor compreensão, nós dividimos esta seção em duas partes, a primeira analisa os resultados vindos dos testes do módulo de Identificação do Usuário e a segunda parte analisa os resultados do módulo de Automação Residencial.

6.3.1 Identificação dos Usuários

A tabela a seguir (Tabela 9) resume os resultados da analise deste módulo.

Requerimento	Resultado
Tempo de inicialização	Satisfatório
Leitura Simultânea de etiquetas RFId	Satisfatório
Tempo para leitura	Satisfatório
Distância em relação a antena	Não Satisfatório
Inclinação da etiqueta RFId	Satisfatório
Acurácia	Satisfatório

Tabela 9 - Resumo dos Resultados

O sistema foi capaz de satisfazer todos os requerimentos com exceção do requerimento relativo a distância a qual a antena é capaz de identificar uma etiqueta RFId.

O *framework* foi capaz de entregar de forma transparente suas funções de controle do equipamento de RFId, que inclui inicialização do leitor, leitura de várias etiquetas simultaneamente e acurácia na identificação do usuário.

O leitor RFId funcionou de acordo com os padrões ao tentar ler uma etiqueta com uma orientação não paralela ao plano da antena. Os valores encontrados foram bem próximos aos padrões, e considerando-se que a precisão dos testes não era tão grande, podemos dizer que antena satisfez os requerimentos.

6.3.2 Análise do Sistema de Automação Residencial

A tabela a seguir (Tabela 10) resume os resultados da analise deste módulo.

Requerimento	Resultado
Tempo para envio de um comando	Satisfatório
Tempo para execução de uma consulta	Satisfatório
Tempo de Inicialização	Satisfatório

Tabela 10 - Resumo dos Resultados

O *framework* foi capaz de se comunicar com o servidor *Bticino* de forma satisfatória em todos os requisitos testados. Um aspecto importante a se enfatizar é que algumas vezes o servidor parece não responder rapidamente ao *framework*, o que nos da a impressão ao usuário que o sistema está congelado. Não foi possível verificar o que estava causando este atraso.

7 CONSIDERAÇÕES FINAIS

Neste tópico procuraremos aprofundar nas questões que nortearam o projeto a fim de concluir se nossos objetivos iniciais foram atingidos e as razões desse nível de sucesso alcançado.

7.1 CONCLUSÃO DO PROJETO

Ambas as tecnologias de RFId e Sistemas *Domóticos* tem crescido em popularidade no contexto mundial, desenvolver um projeto usando tais tecnologias foi uma experiência interessante e nos deu a oportunidade de entender como soluções de computação *pervasiva* podem ser desenhadas e implementadas.

Iniciamos nosso trabalho com a análise da tecnologia RFId, o que nos forneceu o conhecimento necessário para desenvolver uma aplicação baseada em RFId. Os principais resultados e conclusões alcançadas neste projeto foram:

- O *framework* de identificação por RFId alcançou as metas de fornecer um conjunto de funcionalidades transparentes a aplicação *domótica*. As metas relacionadas aos usuários também alcançaram resultados satisfatórios em todos os critérios com única exceção a distancia de detecção que foi considerada insatisfatória;
- O módulo de Automação Residencial foi capaz de fornecer à aplicação a capacidade de comunicação e controle dos dispositivos da casa através do barramento domótico. Os testes de desempenho também alcançaram níveis satisfatórios em todos os critérios, desta vez sem exceção;
- Algumas limitações de determinados equipamentos RFId podem comprometer a *pervasividade* do sistema de identificação humano. Uma pequena inclinação da etiqueta RFId em relação a antena RFId já é capaz de causar a não detecção do bracelete e, portanto, do usuário. Soluções para identificação pervasiva deveriam usar leitores RFId capazes de gerar ondas circulares polarizadas ou etiquetas RFId com antenas tripolares para identificar usuários sem problemas de sensibilidade à inclinação dos braceletes em relação a antena.

O maior problema encontrado no projeto foi o equipamento RFId usado no protótipo. A escolha do equipamento foi limitada pelo baixo custo aspirado pelo projeto Autonomamente. A análise dos testes feitos no sistema de identificação por RFId destacaram a necessidade de um leitor RFId com longo raio de leitura a fim de identificar corretamente os usuários. Porém, como o tamanho e o custo dos equipamentos RFId aumentam proporcionalmente ao raio de leitura, esta solução torna-se inadequada a projetos que buscam pequenas dimensões físicas e baixo custo de produção.

7.2 TRABALHOS FUTUROS

Durante a avaliação do sistema dois importantes pontos de possíveis aprimoramentos apareceram, um relacionado ao módulo de identificação de usuário e o segundo relacionado ao modulo de automação residencial. O tempo médio para identificar o usuário ficou acima do desejado, o que não permitiu ao usuário a sensação que o sistema respondia instantaneamente. É proposto que, no futuro, o adaptador *Serial-Ethernet* utilizado entre o terminal e o leitor de rádio freqüência seja retirado. A primeira razão é que o terminal utilizado durante os testes possuía uma porta serial a qual poderia ser usada para conectar o terminal diretamente ao leitor. A segunda razão é que este conversor acaba por provocar um atraso na comunicação entre os dois dispositivos em questão.

O segundo ponto, referente ao modulo de automação residencial, é que apesar do sistema geralmente responder rapidamente aos comandos enviados pelo terminal, em alguns momentos o sistema congelava segundos antes de realizar o comando. Alguns ajustes relacionados com a fase de conexão entre o sistema e os dispositivos devem ser feitos a fim de estabilizar o tempo de resposta do sistema.

Trabalhos futuros relacionados ao módulo de automação residencial deverão prever o suporte a outros protocolos, como o Konnex, e o uso de arquivos no formato de XML especificando e listando os diversos componentes da casa, como visto em [10].

Outra sugestão é o aumento de integração entre os módulos. Podem-se imaginar cenários diversos, como por exemplo, um onde o módulo de identificação detecta a presença do usuário e então o módulo de automação residencial liga a

câmera e a acende a luz que ilumina a região do terminal. Este tipo de interação poderia incluir suporte a outras antenas localizadas estrategicamente, que acionariam a iluminação do local em que se situa o usuário ou detectaria se usuário se afastou de um fogão, com o objetivo de alertar caso o usuário se esqueça de fechar o gás.

O sistema atualmente é configurado através do uso de arquivos de configuração no formato de XML. A criação de uma interface gráfica que facilite os usuários a alterar de forma segura os parâmetros de configuração. Usuários que tem dificuldades de manipular diretamente nos arquivos de configuração poderão através dessa interface realizar operações como registrar novas pulseiras ou novos componentes da casa.

REFERÊNCIAS

1. Barr, M. Embedded Systems Glossary. Netrino Technical Library. Retrieved on 2007-04-21.

2. Battezzati, L. and Hygounet, J.-L. RFId: Identificazione automatica a radiofrequenza (2nd Edition ed.). HOEPLI, Milan, 2006.

3. Braz, C. and Robert, J. 2006. Security and usability: the case of the user authentication methods. In Proceedings of the 18th international Conferenceof the Association Francophone D'interaction Homme-Machine (Montreal, Canada, April 18 - 21, 2006). IHM '06, vol. 133. ACM Press, New York, NY, 199-203. DOI= http://doi.acm.org/10.1145/1132736.1132768

4. Brummit, B., Meyers, B., Krumm, J., Kern, A. and Shafer S. Easy living: technologies for intelligent environments, in: HUC 2000, (Bristol, UK, September 2000), 1-12

5. Grimm, R. System support for pervasive applications, Ph.D. Thesis, University of Washington, Department of Computer Science and Engineering, December 2002.

6. Miller, R. B. Response time in man-computer conversational transactions. AFIPS Conference Proceedings, Fall 1968, 267-277.

7. Mullen, D. (2006). The Application of RFId Technology in a Port. Retrieved July 12, 2007, from AimGlobal Web Site: http://www.aimglobal.org/technologies/rfid/resources/PortTech.pdf

8. Paradiso, J. and Hsiao, K. Y. 1999. Swept-frequency, magnetically-coupled resonant tags for realtime, continuous, multiparameter control. In CHI '99 Extended Abstracts on Human Factors in Computing Systems (Pittsburgh, Pennsylvania, May 15 - 20, 1999). CHI '99. ACM Press, New York, NY, 212-213. DOI= http://doi.acm.org/10.1145/632716.632848

9. Paret, D. 2005. Technical state of art of "Radio Frequency Identification -- RFId" and implications regarding standardization, regulations, human exposure, privacy. In Proceedings of the 2005 Joint Conference on Smart

Objects and Ambient intelligence: innovative Context-Aware Services: Usages and Technologies (Grenoble, France, October 12 - 14, 2005). sOc-EUSAI '05, vol. 121. ACM Press, New York, NY, 9-11. DOI= http://doi.acm.org/10.1145/1107548.1107555

10. Pellegrino, P., Bonino, D., and Corno, F. 2006. Domotic house gateway. In Proceedings of the 2006 ACM Symposium on Applied Computing (Dijon, France, April 23 - 27, 2006). SAC '06. ACM Press, New York, NY, 1915-1920. DOI= http://doi.acm.org/10.1145/1141277.1141730

11. Perego, A. Le Applicazioni RFId. Politecnico di Milano, Dipartimento di Ingegneria Gestionale, Milan, Not published.

12. Roman, M., Hess, C. and Campbell, M. Gaia: An OO middleware infrastructure for ubiquitous computing environments, in ECOOP Workshop on Object-Orientation and Operating Systems (ECOOPOOOSWS) 2002, Malaga, Spain, June 2002.

13. Römer, K., Schoch, T., Mattern, F. and Dübendorfer, T. 2004. Smart identification frameworks for ubiquitous computing applications. Wirel. Netw. 10, 6 (Nov. 2004), 689-700. DOI= http://dx.doi.org/10.1023/B:WINE.0000044028.20424.85

14. Vahid, F. and Givargis, T. 2002. Embedded Systems Design: A Unified Hardware/Software Introduction. John Wiley & Sons;

15. Verissimo, P., Cahill, V., Casimiro, A., Cheverst, K., Friday, A. and Kaiser, J. CORTEX: Towards supporting autonomous and cooperating sentient entities, in: European Wireless 2002, Florence, Italy (February 2002).

16. Vezzani, R., Cucchiara, R., Grana, C. and Prati, A. Computer vision techniques for PDA accessibility of in-house video surveillance. In First ACM SIGMM international Workshop on Video Surveillance (Berkeley, California, November 02 - 08, 2003). IWVS '03. ACM Press, New York, NY, 87-97. DOI= http://doi.acm.org/10.1145/982452.982464

17. International Organization for Standardization ISO 9241-11: Ergonomic requirements for office work with visual display terminals (VDTs Part 11: Guidance on Usability), 1998.

18. RFId Journal. (2007). Glossary of terms. Retrieved July 12, 2007, from RFId Journal: http://www.rfidjournal.com/article/glossary

19. RFId News. (2004). Understanding anticollision: Processing multiple cards at the same time : RFId News. Retrieved July 12, 2007, from RFId News: http://www.rfidnews.org/library/2004/01/01/understanding-anticollision-processing-multiple-cards-at-the-same-time/

20. School of Management of Politecnico di Milano. 2007. RFId: alla ricerca del valore: Rapporto 2007 Osservatorio RFId. Politecnico di Milano - Dipartimento di Ingegneria Gestionale, Osservatorio RFId. Milano: Politecnico di Milano.

21. Home Page HANDIMATICA. Retrieved April 6, 2007: http://www.handimatica.it

22. Ability Tech Help. Retrieved December 3, 2004: www.ability-tecnhelp.it