



UNIOESTE – Universidade Estadual do Oeste do Paraná

CENTRO DE CIÊNCIAS EXATAS E TECNOLÓGICAS

Colegiado de Ciência da Computação

Curso de Bacharelado em Ciência da Computação

**SACCOB: Sistema de Alerta Contra Colisão com
Obstáculos**

Felipe Schulz Sefrin

CASCVEL

2012

FELIPE SCHULZ SEFRIN

SACCOB: Sistema de Alerta Contra Colisão com Obstáculos

Monografia apresentada como requisito parcial para obtenção do grau de Bacharel em Ciência da Computação, do Centro de Ciências Exatas e Tecnológicas da Universidade Estadual do Oeste do Paraná - Campus de Cascavel

Orientador: Prof. Dr. Jorge Bidarra

CASCADEL

2012

FELIPE SCHULZ SEFRIN

SACCOB: Sistema de Alerta Contra Colisão com Obstáculos

Monografia apresentada como requisito parcial para obtenção do Título de *Bacharel em Ciência da Computação*, pela Universidade Estadual do Oeste do Paraná, Campus de Cascavel, aprovada pela Comissão formada pelos professores:

Prof. Dr. Jorge Bidarra (Orientador)
Colegiado de Ciência da Computação,
UNIOESTE

Prof. M. Eng. Adriana Postal
Colegiado de Ciência da Computação,
UNIOESTE

Prof. Dr. Marcio Seiji Oyamada
Colegiado de Ciência da Computação,
UNIOESTE

Cascavel, 16 de junho de 2012.

DEDICATÓRIA

Dedico este trabalho a todas as pessoas que participaram direta e indiretamente da minha formação acadêmica. Em especial dedico a Deus por ter me proporcionado a vida, aos meus pais Fabiano e Sônia acreditarem na minha capacidade, à minha namorada Lidiane pela compreensão e auxílio que recebi. Ao professor orientador Jorge Bidarra por acreditar e me auxiliar nessa jornada. Aos meus colegas de faculdade que me incentivaram em momentos difíceis.

"A mente que se abre a uma nova ideia, jamais voltará ao seu tamanho original" Einstein.

AGRADECIMENTOS

Primeiramente agradeço a Deus pela saúde e pelo equilíbrio. Aos meus pais, Fabiano e Sônia pelo amor, carinho e confiança depositada em minha capacidade. A minha namorada Lidiane pelo apoio, compreensão, atenção e auxílio em todos os momentos difíceis desta caminhada. Aos meus irmãos Gabriel e Fábio, pelo apoio, amizade e momentos que dividimos juntos, meu muito obrigado. Aos meus colegas de faculdade que estiveram ao meu lado nesta longa jornada e me incentivaram em momentos difíceis, agradeço pelo apoio, convivência e amizade nesses seis anos. Ao professor orientador Jorge Bidarra, pelo apoio, por acreditar e incentivar a realização deste trabalho, meus sinceros agradecimentos. A todas as pessoas que contribuíram diretamente e indiretamente para a realização deste trabalho, meu muito obrigado.

Lista de Figuras

2.1 - Protótipo da Luva Sensorial [15]	09
2.2 - Protótipo do Óculos Sonar [18].....	10
2.3 - Tela de Interface do sistema supervisorio de Junior e Sarno [18]	11
2.4 - Fluxograma do sistema principal proposto por Junior e Sarno. [19]......	12
3.1 - Modelo de <i>Tags</i> [23]	14
3.2 - Modelo de <i>Tag</i> passiva [24]......	15
3.3 - Modelo de <i>Tag</i> semi-passiva [24]	15
3.4 - Modelo de <i>Tag</i> ativa [24].....	16
3.5 - Estrutura do EPC-96 [25]	18
3.6 - Visão de topo da lateração tridimensional	19
3.7 - RFID Middle Range Integrated Reader SAAT-I801 [29].....	20
3.8 - Exemplo de tag utilizada neste sistema [30]......	20
4.1 - Diagrama de Classes do sistema SACCOB	25
4.2 - Exemplo da Sintetização de Voz.....	26
4.3 - Diagrama Entidade Relacional da Base de Dados do sistema SACCOB	27
4.4 - Locais em que a pessoa pode se encontrar em relação a um objeto.....	30
4.5 - Teste de colisão a partir do plano cartesiano.....	31
4.6 - Diagrama Fluxo do sistema SACCOB.....	32
4.7 - Gráfico de tempo de Execução do sistema SACCOB	33
5.1 - Possibilidade de Calcular a rotação de um objeto para este trabalho.....	36
C.1 - E-mail da ECHO XIE referente ao valor do Leitor/Receptor SAAT-I801	45
D.1 - Interface Inicial do sistema SACCOB.....	46
D.2 - Interface Menu Pessoas do sistema SACCOB	47
D.3 - Interface Menu de Objetos do sistema SACCOB	48
D.4 - Interface Menu de <i>Tags</i> do sistema SACCOB.....	48
D.5 - Interface Menu de Locais do sistema SACCOB	49

D.6 - Interface Menu de Receptores do sistema SACCOB.....	49
D.7 - Interface com o botão Iniciar Sistema pressionado.....	50
D.8 - Interface com o Sistema em funcionamento	50

Lista de Tabelas

2.1 - Níveis de Acuidade Visual	05
3.1 - Bandas em que opera o RFID [24].....	17
5.1 - Custo estimado dos equipamentos utilizados no desenvolvimento do sistema	34

Lista de Abreviaturas e Siglas

API	<i>Application Programming Interface</i>
AV	Acuidade Visual
CPU	<i>Central Processing Unit</i>
DAO	<i>Data Access Object</i>
EPC	<i>Electronic Product Code</i>
FINEP	Financiadora de Estudos e Projetos
IBGE	Instituto Brasileiro de Geografia e Estatística
ICIDH	<i>International Classification of Impairments, Disabilities and Handicaps</i>
ID	Identificador
ISO	<i>International Organization for Standardization</i>
MVC	<i>Model View Controller</i>
OMS	Organização Mundial de Saúde
ONU	Organização das Nações Unidas
RFID	Identificação e Rastreamento por Rádio Frequência
RW	<i>Read/Write</i>
SAR	Sistema de Automação Residencial
SCAA	Sistema de Controle Automático de Ambientes
SDK	<i>Software Developers Kit</i>
SGBD	Sistema de Gerenciamento de Banco de Dados
SUS	Sistema Único de Saúde
TA	Tecnologia Assistiva

WORM *Write Once Read Many*

XP *Extreme Programming*

Sumário

Lista de Figuras	vi
Lista de Tabelas	viii
Lista de Abreviaturas e Siglas	ix
Sumário.....	xi
Resumo	xiii
1 Introdução.....	1
1.1 Objeto de Estudo.....	2
1.2 Proposta	2
1.3 Organização do Documento	2
2 Pessoas com Deficiência: Problemas e Soluções	4
2.1 Conceitos de Deficiência	4
2.2 Classificações de Deficiências	4
2.2.1 Deficiência Visual	5
2.3 Investimentos do Governo Brasileiro no Campo do Desenvolvimento das Tecnologias Assistivas	6
2.4 O papel das TAs para pessoas com deficiência	7
2.5 TAs voltadas a Deficientes Visuais	8
2.6 Implementação Correlata.....	10
3 Enquadramento Tecnológico.....	13
3.1 Domótica.....	13
3.2 Sistema RFID.....	14
3.2.1 Componentes do RFID	14
3.2.2 Memórias das tags	16
3.2.3 Funcionamento do RFID	16
3.2.4 Desafios do RFID	17
3.2.5 Interferências no RFID	18

3.3 Triangulação	19
3.4 Sobre a Simulação nesse Trabalho	20
4 O Sistema Proposto	21
4.1 Visão do Sistema SACCOB	21
4.2 Modelagem	22
4.2.1 Requisitos Funcionais	22
4.2.2 Requisitos Não Funcionais	23
4.2.3 Diagrama de Classes	24
4.3 Sintetização de Voz	26
4.4 Base de dados.....	27
4.4.1 Diagrama Entidade Relacional da Base de Dados.....	27
4.5 Módulos de Leitura de Tags e Teste de Colisão.....	28
4.5.1 Módulo de Leitura de Tags.....	28
4.5.2 Módulo de Teste de Colisão	30
4.6 Funcionamento.....	31
4.7 Análises dos Resultados.....	33
5 Conclusões	34
5.1 Considerações Finais	34
5.2 Perspectivas Futuras	35
Apêndice A	37
Apêndice B	40
Apêndice C	45
Apêndice D	46
Referências Bibliográficas	51

Resumo

Este trabalho pretende proporcionar às pessoas com deficiência visual uma maior segurança em sua movimentação, reduzir o índice de colisões contra obstáculos e aumentar a liberdade de locomoção. Por meio do desenvolvimento de um sistema computacional, na área de Domótica, batizado de SACCOB (Sistema de Alerta Contra Colisão com Obstáculos), baseando-se em resumos que tenham sido escritos/publicados nesta mesma área de desenvolvimento, que a partir de sensores dispostos em cada obstáculo e também no próprio indivíduo, seja capaz de calcular a distância entre o obstáculo e o deficiente visual, avisando-o sobre a sua proximidade. Para que fosse possível mapear a pessoa com deficiência visual e os obstáculos no ambiente foi utilizada a tecnologia RFID (*Radio Frequency Identification*), que presta suporte ao algoritmo de triangulação utilizando a técnica de lateração para encontrar as posições. Logo que se realiza o mapeamento, trabalha-se com as posições para verificar e evitar colisões contra os obstáculos, mas de acordo com o desenvolvimento são possíveis nove posições entre o indivíduo mapeado e os obstáculos. Sendo assim, com as posições estabelecidas o sistema tem condições de encontrar o ponto do obstáculo mais próximo ao indivíduo e realizar testes a partir da técnica de distância entre dois pontos. Caso o teste verifique que essa distância é menor que a distância mínima, é emitido um aviso ao usuário utilizando sintetização de voz. Foram efetuados testes em diversos tipos de cenários, e o sistema mostrou uma redução de 86% nas colisões sofridas pela pessoa com deficiência visual.

Palavras-chave: Deficiência Visual, Colisão contra Obstáculos, RFID.

Capítulo 1

Introdução

A Organização Mundial de Saúde (OMS) estima que existam cerca de 285 milhões de pessoas com deficiência visual no mundo, das quais 39 milhões são indivíduos cegos e 246 milhões têm baixa visão [1]. No Brasil, segundo o Censo de 2010 [2], publicado pelo Instituto Brasileiro de Geografia e Estatística (IBGE) mostra que 35,7 milhões de pessoas são deficientes visuais; quantidade essa que vem aumentando significativamente ao longo dos anos, comparado ao Censo 2000 [3], onde apenas 16,6 milhões de pessoas eram consideradas deficientes visuais.

Tanto no Brasil como em outros países, vários tipos de soluções tecnológicas assistivas vêm sendo desenvolvidas por empresas e também por institutos de pesquisa com o objetivo de auxiliarem esses indivíduos na realização de suas atividades diárias. Dentre as tecnologias mais utilizadas recentemente, encontramos sistemas de reprodução em braile, ampliadores e leitores de tela de computadores, bengalas sinalizadoras de obstáculos, passarelas com trilhas específicas para a movimentação de cegos em calçadas ou ruas e sistemas de controle de ambiente, onde o objetivo principal destas tecnologias é garantir às pessoas com deficiência visual a acessibilidade [4].

Define-se a acessibilidade, segundo o Decreto nº 5.296 [5] de 2 de dezembro de 2004, como sendo “possibilidade e condição de alcance para utilização, com segurança e autonomia, dos espaços, mobiliários e equipamentos urbanos, das edificações, dos transportes e dos sistemas e meios de comunicação, por pessoa portadora de deficiência ou com mobilidade reduzida”. Além do citado, acessibilidade também se direciona a eliminação de barreiras envolvendo preconceito e discriminação referentes às necessidades e potencialidades das pessoas com deficiência [6].

Com os recursos tecnológicos disponíveis na área de computação, propõem-se desenvolver um sistema autônomo, cuja atuação será o monitoramento e o fornecimento de informações sonoras, minimizando a colisão de um indivíduo que possui deficiência visual no ambiente

controlado. Com este sistema, o indivíduo que possui deficiência visual pode se locomover com maior facilidade, sem se preocupar com objetos fora do lugar. Portanto, este trabalho demonstra a realização de um processo de reabilitação, retornando a autoestima e inclusão social ao usuário deste sistema, onde o mesmo poderá se locomover livremente, sem depender de um acompanhante para auxiliá-lo.

1.1 Objeto de Estudo

Diante da dificuldade de locomoção de pessoas portadoras de deficiências visuais nos ambientes, sejam eles em suas casas, nas ruas, nos locais de trabalho e assim por diante, um problema desafiador se coloca: como tentar minimizar as dificuldades dessas pessoas quando em circulação por esses espaços? Partindo-se desse ponto, o objeto de estudo que aqui se desenvolve é a implementação de um sistema que controle o ambiente e interaja com o usuário, a partir de alertas sonoros, que informarão o usuário caso o mesmo esteja próximo de se colidir com um obstáculo. Este controle deve ser feito, diferenciando a pessoa mapeada das demais e de obstáculos (objetos) que estarão sendo monitorados.

1.2 Proposta

Buscando contribuir com área de desenvolvimento de Sistemas de Controle Automático de Ambientes (SCAA), especialmente no que diz respeito à domótica, propõe-se aqui modelar e implementar um protótipo de sistema de controle automático de ambientes, baseado em alertas que utilizam a programação concorrente. O sistema terá que mapear o usuário em todo o ambiente controlado, auxiliando-o em seu deslocamento pelos obstáculos (objetos mapeados a partir de sensores posicionados nos mesmos), reproduzindo frases na língua portuguesa em cada aviso emitido.

1.3 Organização do Documento

Em termos estruturais, esse trabalho assim se organiza. No capítulo 1, encontramos uma introdução, que faz uma apresentação sobre o contexto em que um indivíduo com deficiência visual está inserido e apresentaremos o objetivo e solução proposta de nosso trabalho.

No Capítulo 2 apresentamos alguns conceitos, dividido em duas seções: a primeira apresenta os conceitos de tecnologias assistivas; já a segunda seção conceitua e classifica as deficiências com ênfase na deficiência visual. Além disto, discutiremos alguns conceitos de tecnologias assistivas, como podemos aplicá-las, e exemplos de TAs (Tecnologias Assistivas) voltadas para deficientes visuais.

No Capítulo 3 discutiremos sobre o enquadramento tecnológico referente a este trabalho, discutindo e apresentando sobre domótica, sobre a tecnologia de sensores e forma de triangulação.

O Capítulo 4 é dedicado à apresentação do trabalho em si, sendo descrito a especificação, modelagem e implementação juntamente com uma análise dos resultados obtidos. Teremos uma visão detalhada de cada atividade realizada dentro do procedimento de desenvolvimento do mesmo, onde serão abordadas importantes etapas do processo. Sendo assim, podemos destacar as estruturas: Módulo de Leitura de *Tags* e o Módulo de Teste de Colisão.

Por fim, no capítulo 5 apresentaremos as conclusões referentes ao nosso trabalho, e as possibilidades da realização de trabalhos futuros a partir destes estudos.

Capítulo 2

Pessoas com Deficiência: Problemas e Soluções

2.1 Conceitos de Deficiências

Segundo o Instituto Benjamin Constant [6], um indivíduo é considerado portador de deficiência quando o mesmo apresenta, em caráter permanente, perdas ou reduções em sua estrutura física ou função anatômica, fisiológica, psicológica ou mental, limitando-os para a realização de algum tipo de atividade ou tarefa em suas vidas diárias.

Em 1993, a Organização Mundial da Saúde (OMS) lançou a 3ª Classificação Internacional das Deficiências, Incapacidades e Desvantagens [7] (3ª *International Classification of Impairments, Disabilities and Handicaps* – ICIDH). De acordo com a OMS, as deficiências foram classificadas com base em três conceitos chaves, a saber:

- Deficiência: é a perda ou anormalidade da estrutura psicológica, ou fisiológica, anatômica ou uma função.
- Incapacidade: consideramos qualquer restrição ou falta de capacidade do indivíduo realizar uma atividade de forma normal para o ser humano.
- Desvantagem: considerado uma desvantagem para um indivíduo, resultante de uma deficiência ou incapacidade, cujo limite impeça o mesmo de cumprir um papel que é normal para aquele indivíduo.

2.2 Classificações de Deficiências

De acordo com os conceitos do ICIDH, a OMS classifica as deficiências como sendo: deficiência mental, deficiência física, deficiência auditiva, deficiência múltipla e deficiência visual [7].

O Instituto Benjamin Constant [6] classifica as deficiências em: deficiência motora, deficiência cerebral, deficiência mental, deficiência visual, deficiência auditiva e deficiência múltipla. No entanto, neste trabalho enfatizaremos a deficiência visual, com o intuito de proporcionar uma ferramenta para a inserção dessas pessoas na sociedade.

2.2.1 Deficiência Visual

A deficiência visual é a redução ou perda da capacidade visual onde não se tem uma melhora ou correção através do uso de lentes, de tratamento clínico ou cirúrgico, como apresenta o Instituto Benjamin Constant [6].

Acuidade Visual (AV) é o grau de aptidão do olho, ou seja, a capacidade de percepção na forma e no contorno de objetos. Essa capacidade é atribuída aos cones (células fotossensíveis presentes na retina), que se responsabilizam pela Acuidade Visual central, compreendendo a visão de forma e a visão de cores [8]. Na tabela 2.1 apresentamos os níveis de AV.

Tabela 2.1 – Níveis de Acuidade Visual [8]

Classificação	Acuidade Visual de Snellen	Acuidade Visual Decimal	Auxílios
Visão Normal	20/12 a 20/25	1,5 a 0,8	<ul style="list-style-type: none"> • Bifocais comuns
Próximo do normal	20/30 a 20/60	0,6 a 0,3	<ul style="list-style-type: none"> • Bifocais mais fortes • Lupas de baixo poder
Baixa visão moderada	20/80 a 20/150	0,25 a 0,12	<ul style="list-style-type: none"> • Lentes esferoprismáticas • Lupas mais fortes
Baixa visão profunda	20/500 a 20/1000	0,04 a 0,02	<ul style="list-style-type: none"> • Lupa montada telescópio • Magnificação vídeo • Bengala • Treinamento Orientação/Mobilidade
Próximo à cegueira	20/1200 a 20/2500	0,015 a 0,008	<ul style="list-style-type: none"> • Magnificação vídeo livros falados, Braille • Aparelhos de saída de voz • Softwares com sintetizadores de voz • Bengala • Treinamento Orientação/Mobilidade
Cegueira total	Sem projeção de luz	Sem projeção de luz	<ul style="list-style-type: none"> • Aparelhos de saída de voz • Softwares com sintetizadores de voz • Bengala • Treinamento Orientação/Mobilidade

Conforme Antônio Menescal Conde [6], professor do Instituto Benjamin Constant, em 1966 a OMS registrou 66 definições para cegueira. Para que este número fosse simplificado,

em 1972 a OMS propôs normas para a definição de cegueira e para uniformizar as anotações dos valores de acuidade visual com finalidades estatísticas.

Para cegueira parcial, os indivíduos são capazes apenas de enxergar e perceber vultos (contornos sinuosos) a uma curta distância (como aproximar os dedos aos olhos e conta-los). Como se encontram muito próximos da cegueira total, estes indivíduos possuem apenas a percepção (distinção entre claro e escuro) e projeção de luz (capaz de identificar também a direção de onde provém a luz). Já a cegueira total é a perda completa de visão, onde a visão é totalmente nula, isto é, nem a projeção luminosa está presente.

Um indivíduo é considerado cego quando a visão corrigida do melhor dos seus olhos é de 20/200 ou menos, isto é, a mesma consegue enxergar a 20 pés (6 metros) e uma pessoa de visão normal pode enxergar a 200 pés (60 metros), ou se o diâmetro mais largo do seu campo visual forma um arco menor que 50 graus, ainda que sua AV nesse estreito campo possa ser superior a 20/200. Caracteriza-se então, como portador de visão subnormal aquele que possui acuidade visual de 6/60 e 18/60 e/ou um campo visual entre 20 e 50°.

Portanto, é tratado como cego o indivíduo que necessita de instrução em Braille, e como portador de visão subnormal aquele que lê tipos impressos ampliados ou com o auxílio de recursos ou ferramentas ópticas.

2.3 Investimentos do Governo Brasileiro no Campo do Desenvolvimento das Tecnologias Assistivas

O Ministério da Saúde publica uma cartilha que apresenta o resultado de diversas investigações de usuários e seus familiares sobre ações de acessibilidade, prevenção de deficiências, assistência à saúde da pessoa com deficiência e processos de reabilitação. Incluindo a isto, a dispensação de órteses, próteses, meios auxiliares de locomoção e bolsas de ostomia, fornecidos pelas unidades de saúde que integram a rede de serviços do Sistema Único de Saúde (SUS), em todo o País [9].

A Política Nacional de Saúde da Pessoa com Deficiência (Portaria MS/GM nº 1.060, de 5 (cinco) de junho de 2002 [10]) define uma gama de possibilidades, seguindo da prevenção de agravos à saúde até a reabilitação de pessoas com deficiência em sua capacidade funcional e desempenho, auxiliando sua inclusão em todas as esferas sociais. Um dos pilares da política tem em foco o fortalecimento de movimentos institucionais, que buscam ampliar a

acessibilidade e inclusão. Desta forma, visa também a melhoria de acesso a estruturas físicas, a informações e aos benefícios e serviços que se encontram disponíveis às pessoas com deficiência no SUS [11].

Segundo a Organização das Nações Unidas (ONU), o Brasil encontra-se entre os 35% dos países membros da ONU que disponibilizam uma legislação para as pessoas com deficiência. O país caracteriza-se como atuador nas áreas de direitos humanos; na defesa de valores de dignidade, inclusão e acessibilidade; na melhoria das condições de vida; no acesso a ambientes e serviços públicos tais como educação, saúde, transporte e segurança. Foram apresentados avanços na legislação em relação à educação inclusiva, atenção à saúde, a reabilitação e seus complementos, como por exemplo, as órteses, próteses, meio de locomoção e cotas para inclusão no mercado de trabalho [9].

Cada vez mais o Brasil vem investindo em processo de acessibilidade para cada um dos tipos de deficiência, sendo elas, adaptações nos ambientes físicos e meios de transporte, em formas para comunicação de informações às pessoas com deficiência auditiva ou visual e implementação de políticas de ações afirmativas entre outras áreas envolvidas [9].

A Financiadora de Estudos e Projetos (FINEP) apresentou uma chamada pública no dia 03 de janeiro de 2012, onde financiará projetos em Tecnologia Assistiva com o investimento de R\$ 20 milhões. Estes recursos se direcionam para o desenvolvimento de produtos, metodologias, estratégias, práticas e serviços que aumentem a autonomia e a qualidade de vida de idosos e pessoas com deficiência. Os mesmos se integram ao investimento do Plano Nacional dos Direitos das Pessoas com Deficiência - Viver sem Limite - lançado pelo Governo Federal em 17 de novembro 2011 [12].

Com base na existência das diversas deficiências, estes projetos irão abranger o desenvolvimento de Softwares de reconhecimento de voz, impressoras Braille, tecnologia nanoeletrônica aplicada a aparelhos auditivos e próteses para articulações, que estarão presentes na sociedade como um meio de inserção [12].

2.4 O papel das TAs para pessoas com deficiência

Bersh [4] define Tecnologia Assistiva como um arsenal de recursos e serviços que possam contribuir e auxiliar a pessoa portadora de algum tipo de deficiência, para o acréscimo de suas habilidades funcionais, promovendo assim uma melhora em sua qualidade de vida.

Segundo a autora [4], as TA se classificam da seguinte maneira: TAs para Auxílios na vida diária e vida prática; Comunicação Aumentativa e Alternativa; Recursos de acessibilidade ao computador; Sistemas de controle de ambiente; Projetos arquitetônicos para acessibilidade; Órteses e próteses; Adequação Postural; Auxílios de mobilidade; Auxílios para cegos ou para pessoas com visão subnormal; Auxílios para pessoas com surdez ou com déficit auditivo e Adaptações em veículos.

Dentre os tipos de TAs, nosso trabalho está focado na categoria de Sistema de controle de ambiente. Estes sistemas de controle de ambiente podem fazer o mapeamento de locais, podendo ser um quarto, um escritório e até mesmo uma residência. Dependendo de sua funcionalidade, pode auxiliar pessoas com deficiência motora, facilitando o acionamento ou desligamento de aparelhos eletrônicos ou de sistema de segurança. No entanto, pode auxiliar pessoas com deficiência visual em sua locomoção, instruir uma navegação facilitada, assim como as outras funcionalidades tratadas para pessoas com deficiência motora [4].

2.5 TAs voltadas a Deficientes Visuais

Existem diversas TAs para o auxílio à pessoas com deficiência visual. Serão mostrados alguns tipos de TAs com um breve resumo descritivo de cada uma delas.

Leitores de Tela: são sistemas que processam um texto que está presente na tela do computador e o reproduz em forma de sintetização de voz ou em um display braile (dispositivo que apresenta pontos, onde os mesmos podem ser salientados ou rebaixados, permitindo a leitura por um deficiente visual). Alguns exemplos do mesmo são: o JAWS [13] e o DOSVOX [14], que possui um sistema completo dotado de leitor de tela.

Luva Sensorial: é uma luva com sensores térmicos dispostos homogeneamente, apresentada na figura 2.1, que são estimulados com o aumento da temperatura fazendo com que um atuador ligado a esta luva, vibre mais intensamente com o aumento da temperatura. Com esta luva podemos evitar que um deficiente visual exponha a sua pele a altas temperaturas, como exemplo, retirar uma panela do fogo, onde o deficiente deveria se preocupar com partes muito quentes do metal [15].

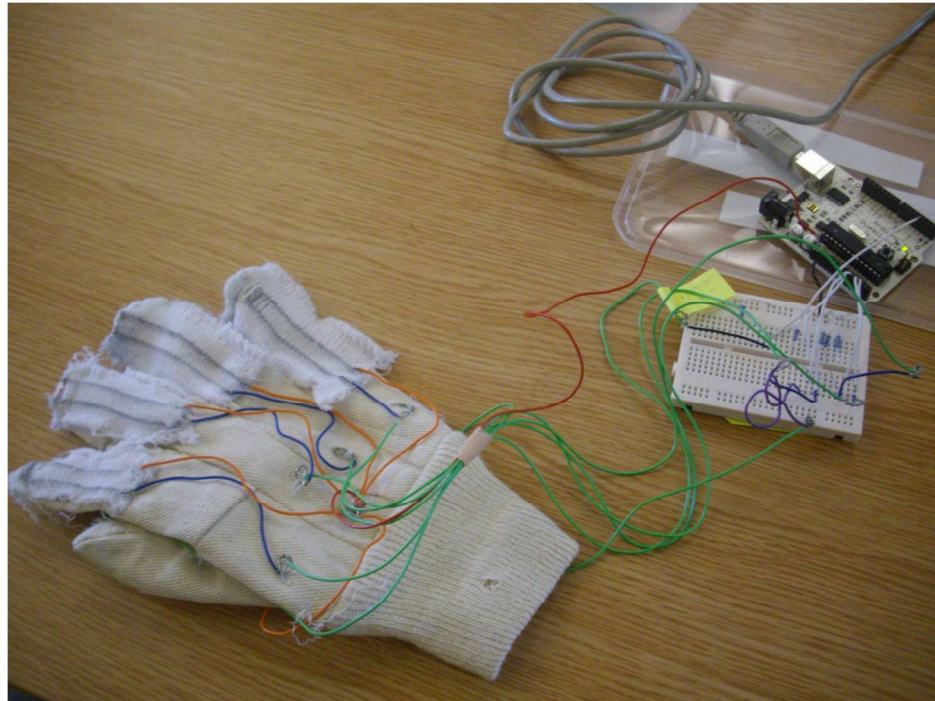


Figura 2.1 - Protótipo da Luva Sensorial [15].

Bengala Eletrônica: É uma bengala eletrônica baseada em sensores de ultrassom para detectar obstáculos e determinar a distância do mesmo. Esta bengala pode identificar obstáculos a uma distância de 0,15 metros a 6,45 metros, e a resposta fornecida para o deficiente visual é realizada através do tato, em forma de vibração que varia de acordo com a distância do obstáculo [16].

Segundo estudos realizados por Celestino e Abe [17] em relação à utilização de bengalas eletrônicas, junto a voluntários que possuem deficiência visual, observaram que havia uma grande preocupação em proteger a parte superior do corpo como o rosto e os ombros. O uso de uma bengala eletrônica proporciona uma varredura eficiente, mas se limita em relação à cobertura da parte superior do corpo. A bengala conseguiu detectar apenas objetos parados, não auxiliando ao deficiente visual uma percepção dos objetos que se encontram em movimento.

Óculos Sonar: Óculos sonar com a utilização de um sensor ultrassônico, que realiza a interface das informações no ambiente, possibilitando a determinação de obstáculos a uma distância de 0,20 metros a 1,50 metros. Para interação com o usuário, utilizou-se um pequeno alto-falante com um potenciômetro que quanto mais próximo do obstáculo, maior é a

frequência do sinal sonoro emitido em um fone de ouvido, que o usuário estará usando [18]. Na figura 2.2 mostramos um protótipo destes óculos.

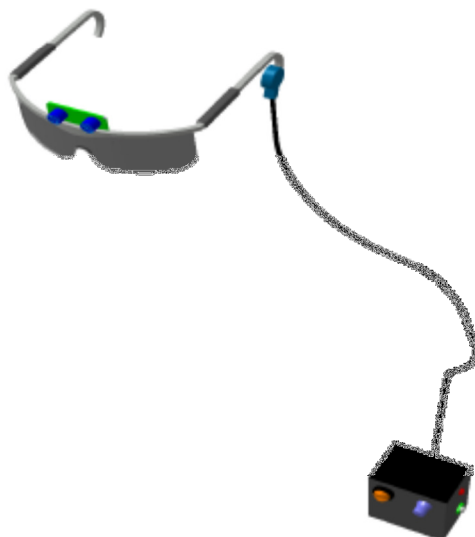


Figura 2.2 – Protótipo do Óculos Sonar [18].

2.6 Implementação Correlata

Junior e Sarno [19] apresentam um sistema para auxiliar as pessoas com deficiência visual a se locomoverem em ambientes limitados ou desconhecidos sem grandes esforços ou dificuldades. Citamos alguns ambientes que podem apresentar essas características, sendo quartos de hotéis, hospitais entre muitas outras possibilidades. Eles utilizam de tecnologias baseadas em microcontroladores e multiplexadores, com a capacidade de mapear, por meio de chave digital, a pessoa com deficiência visual, a partir dos pisos pressionados (cada qual com um sensor correspondente) pela mesma, conforme visualizamos na figura 2.3.

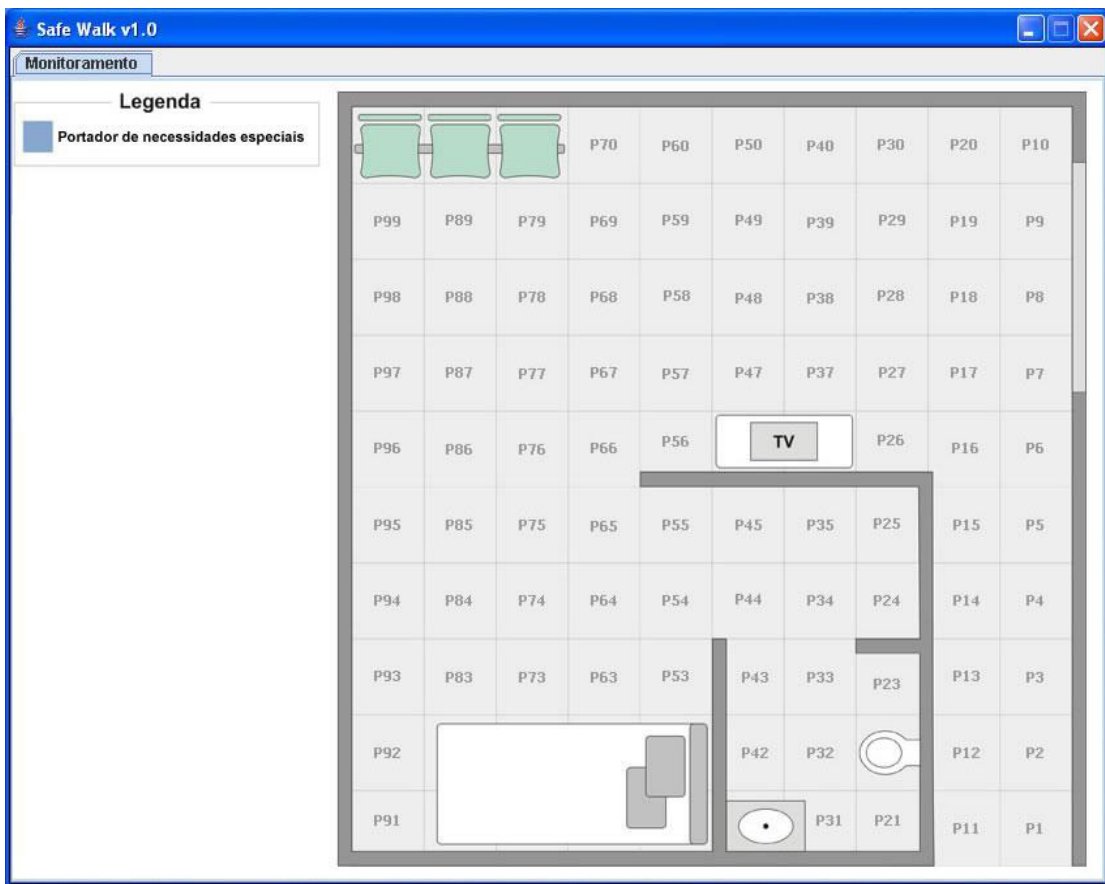


Figura 2.3 - Tela da Interface do sistema supervisorio de Junior e Sarno [19].

Para o módulo de mapeamento, implementou-se um programa que residisse na memória interna do microcontrolador para que controle todo o mesmo. Foi desenvolvido um sistema com a função de varrer continuamente a matriz de micro-switches, enviando sinal de cada um deles que esteja pressionado. Na figura 2.4 podemos verificar através do fluxograma o funcionamento do sistema principal apresentado por Junior e Sarno [19].

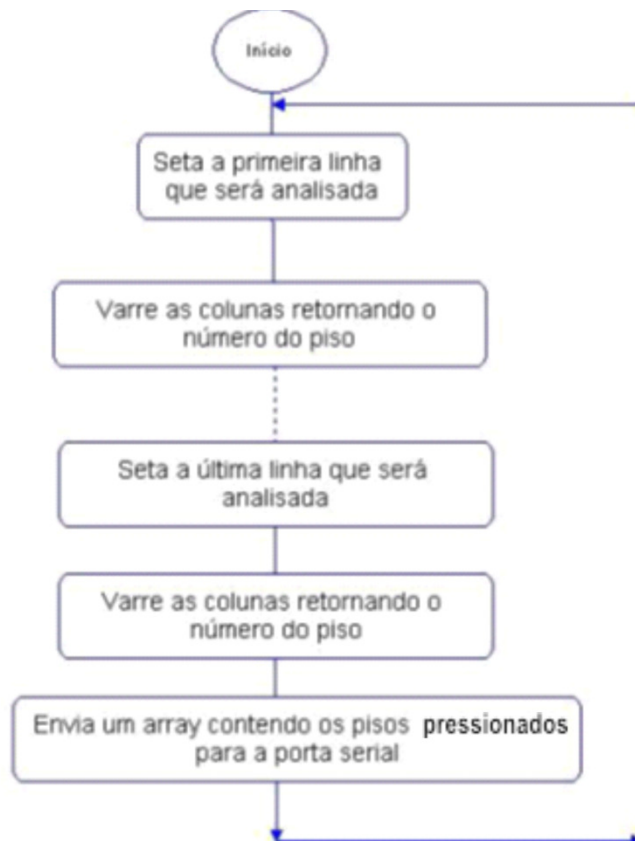


Figura 2.4 - Fluxograma do sistema principal proposto por Junior e Sarno. [19].

A maior dificuldade do sistema implementado por Junior e Sarno [19] foi com relação à comunicação, configuração e instalação da API JavaCom, por falta de documentação, o que acarretou falhas entre a sincronização do módulo de mapeamento e o sistema supervisor, gerando leituras de valores indesejados pela porta serial. Além disso, ocorria a perda de sinal quando mais de um piso era pressionado em uma mesma linha, e inconsistências do sinal de saída do multiplexador (controlador de pressionamento dos pisos) quando nenhum piso estava sendo pressionado.

Capítulo 3

Enquadramento Tecnológico

Sistema de Automação Residencial (SAR) é a utilização de tecnologia que facilita e torna automática tarefas habituais que seriam de responsabilidade do morador. Utilizando sensores de presença, temporizadores ou pressionando um botão de um controle remoto, torna-se possível o acionamento de tarefas, que podem trazer maior praticidade, segurança, economia e conforto para o morador [20].

Os primeiros SARs foram construídos com a finalidade de atender aplicações direcionadas a indústrias na década de 70. Logo em seguida, o comércio decidiu utilizar estes sistemas de automação e vem evoluindo os mesmos até os dias atuais, contando com a participação do rápido avanço da informática e os softwares de supervisão que apresentam aspectos de grande sofisticação. Lojas de departamento, supermercados, hotéis, entre outros, possuem suas operações totalmente integradas, juntamente com sua logística, vendas, finanças, entre outros setores [21].

Um dos principais focos no desenvolvimento de SAR está no gerenciamento de recursos, conforto, segurança e entretenimento. Podemos dizer que estes SAR fazem parte da Domótica [22].

3.1 Domótica

Entende-se domótica como sendo o controle e automação de residências, tendo objetivos em apresentar um maior conforto e uma maior segurança, tanto em nível da detecção de situações de emergência, como incêndios, falhas de segurança, quanto em nível da detecção e sinalização de situações, como localizar objetos. Temos uma série de possibilidades, tratando-se de conforto, como por exemplo, o controle de luminosidade residencial, controle térmico do ambiente e ligar e desligar equipamentos eletrônicos [22].

Neste contexto, utilizamos dos conceitos de domótica como um auxílio no desenvolvimento de uma solução viável para o problema apresentado neste trabalho.

3.2 Sistema RFID

Diferente da tecnologia apresentada por Junior e Sarno, a chamada RFID (*Radio Frequency Identification*) é uma tecnologia que possibilita muitos benefícios para indústrias e outros ramos comerciais. Segundo Gomes [24] a vantagem de utilizar o RFID é permitir a identificação automática de pessoas e objetos a distâncias consideráveis, através de ondas de rádio.

3.2.1 Componentes do RFID

O RFID consiste em três partes, as *tags*, os Leitores/Receptores e o software que fará o processamento dos dados provenientes dos Leitores/Receptores.

Os Leitores/Receptores são dispositivos que consistem em um transceptor de radiofrequência que opera em uma faixa padronizada de frequência, emitindo sinais de rádio, e realiza a leitura/decodificação de informações proveniente das respostas das *tags* [23].

As *tags* usualmente são utilizadas em produtos, como uma etiqueta adesiva de código de barras, mas podemos coloca-las em encapsulamentos resistentes, em cartões de identificação ou até mesmo pulseiras como mostra a figura 3.1. As *tags* podem ser divididas em três grupos: *tags* passivas, *tags* semi-passivas e *tags* ativas [23].

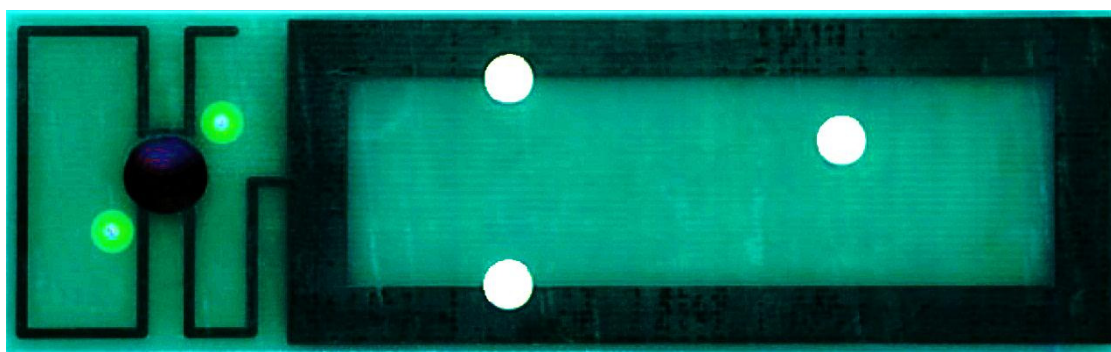


Figura 3.1 – Modelo de *Tag* [23].

Tags passivas: são *tags* que não possuem fonte própria de energia, portanto, ao receberem o sinal de radiofrequência do Leitor/Receptor, parte dessa energia é transformada em corrente

elétrica dentro da *tag*, que por sua vez gerará um sinal de resposta apenas com os dados de identificação contidos em sua memória Prom. Com isto, temos a vantagem de que a *tag* tem um custo mais baixo, mas exigindo um Leitor/Receptor de maior potência, pois uma fração da energia irradiada pelo mesmo é realmente utilizada na resposta da *tag*. Um exemplo da mesma é apresentado na figura 3.2.



Figura 3.2 – Modelo de *Tag* passiva [24].

Tags semi-passivas: são *tags* que são ativadas ao receber um sinal do Leitor/Receptor, que possuem uma bateria interna responsável por alimentar o chip que as *tags* possuem. Estas *tags* continuam utilizando a energia recebida do Leitor/Receptor para realizar a comunicação, porém conseguem armazenar mais dados em suas memórias. Um exemplo da mesma é apresentado na figura 3.3.



Figura 3.3 – Modelo de *Tag* semi-passiva [24].

Tags ativas: são *tags* que possuem uma bateria para alimentar o chip e a comunicação com o Leitor/Receptor. Portanto, esta *tag* consegue emitir continuamente seu sinal, independente do recebimento de um sinal do Leitor/Receptor. Possuem um custo mais alto, porém podem ser identificadas a distâncias maiores, e possuem maiores recursos. Um exemplo da mesma é apresentado na figura 3.4.

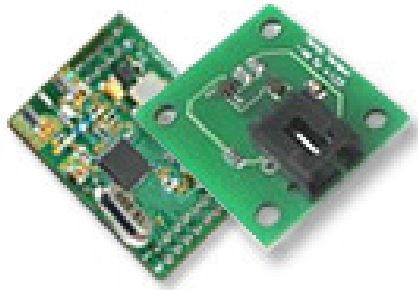


Figura 3.4 – Modelo de *Tag* ativa [24].

Podemos encontrar diversos tipos de dados nas *tags*, entre eles, números de série, instruções para configuração, os histórico de suas atividades (como data da última manutenção, o momento em que a *tag* passou por um local específico, temperatura, dentre outros). [23].

3.2.2 Memórias das *tags*

Uma *tag* pode ser do tipo WORM (*Write Once Read Many*) ou RW (*read/write*). Nas *tags* WORM, a informação referente à identificação da *tag* fica permanentemente em sua memória, nas *tags* RW este número pode ser parcialmente alterado. As *tags* passivas caracterizam-se por ter uma memória não volátil que pode ir desde os 64 bits a 1 kbyte, e as *tags* ativas possuem memórias maiores que podem ser de 16 bytes aos 128 kbytes [26].

3.2.3 Funcionamento do RFID

O funcionamento da RFID acontece quando um Leitor/Receptor emite uma onda de rádio frequência. A *tag* por sua vez, ao receber este sinal proveniente do Leitor/Receptor, responde emitindo um sinal que é identificado e decodificado pelo Leitor/Receptor, que logo em seguida, transfere os dados para o sistema de computador [23].

3.2.4 Desafios do RFID

Na tabela 3.1 são apresentadas as diferentes bandas que o RFID opera, podendo observar as características técnicas e as principais aplicações de cada uma delas.

Tabela 3.1 – Bandas em que opera o RFID [25]

Bandas de frequência	LF Low Frequency	HF High Frequency	UHF Ultra High Frequency	Microwaves
Frequências	30 - 30 kHz	3 - 30 MHz	300 MHz - 3 GHz	1 - 30 GHz
Frequências típicas RFID	125 - 134 kHz	13,56 MHz	433 MHz 865 - 956 MHz 2,45 GHz	2,45 GHz
Alcance Aproximado	< 0,5 m	Até 1,5 m	433 MHz-Até 100 m 865-956 MHz-0,5 a 3 m	Até 10 m (indoor)
Débito Binário	< 1 Kbps	aprox. 25 Kbps	433-956 MHz-30 Kbps 2,45 GHz-100 Kbps	Até 100 Kbps
Características	Curto alcance, baixo débito binário, penetra na água e metal	Maiores alcances, débito binário razoável semelhante a GSM	Longo alcance, elevado débito binário, leitura de até 100 itens simultâneos	Longo alcance elevado débito, não penetra água e metal
Uso típico	Identificação animal, mobilizador	Etiquetas inteligentes, acesso e segurança	Monitorização e localização em animais e logística	Portagens em auto-estradas

Neste trabalho, utiliza-se a *Ultra High Frequency*, com 865-956MHz, onde o alcance pode chegar a mais de 5 metros.

Foram simulados Leitores/Receptores que utilizam o padrão de comunicação ISO 18000-6B [27] e o padrão EPC-96 [25] que vem a ser o número único de identificação para cada *tag* RFID que é guardado em sua memória. Esse sistema de codificação *Auto-ID Centre* consiste em um código com 96 bits, separados em quatro campos: *Header*, *EPC Manager*, *Object Class* e *Serial Number*, como ilustrado na figura 3.5.

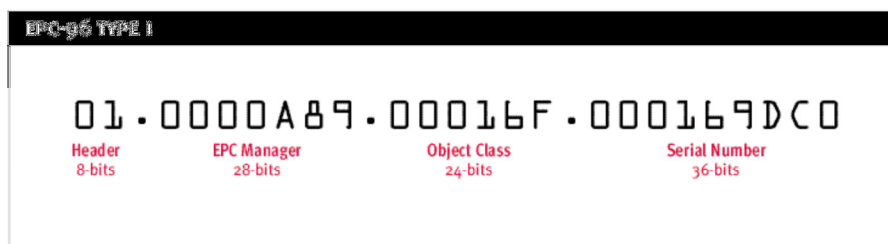


Figura 3.5 – Estrutura do EPC-96 [25].

A estrutura do EPC pode conter de 64 até 256 bits, portanto, na figura 3.5 está sendo exemplificado o EPC do tipo 1 que contém 96 bits (padrão utilizado neste trabalho). O *Header* possui 8 bits e indica o tipo do EPC. No exemplo da figura 3.3, “01” indica que é um EPC do tipo 1. O campo *EPC Manager* contém o número de identificação do fabricante do objeto ao qual a *tag* está associada. Com este campo de 28 bits permite a geração de 268 milhões de códigos únicos para os fabricantes. No campo *Object Class* é identificado o tipo do produto etiquetado. Num código com 96 bits, cada fabricante pode ter 16 milhões de produtos. O *Serial Number* é composto por 36 bits. É nesse campo que está registrada a identificação do objeto monitorado [25].

3.2.5 Interferências no RFID

Ao se utilizar RFID, depara-se com ondas de frequências de rádio, que por sua vez podem sofrer diferentes tipos de interferência. Essas interferências podem ser consideradas internas ou externas.

As interferências internas são as colisões entre leitores, onde dois leitores/receptores fazem leitura em uma mesma *tag*, e a mesma não consegue responder a mais de uma leitura simultaneamente; ou quando é feita a leitura de múltiplas *tags* se encontram em um lugar relativamente pequeno [28].

Por sua vez, as interferências externas, oriundas de materiais que podem realizar interferência por trabalharem com a mesma frequência que os leitores, ou matérias que bloqueiem os sinais emitidos pelos leitores, como as paredes [28].

Neste trabalho não estão sendo consideradas interferências externas, apenas as interferências internas.

Para solucionar estas interferências, é ativado apenas um Leitor/Receptor por vez, onde o mesmo irá capturar as informações de todas as *tags* do ambiente, e a concentração de *tags* será pequena, não tendo o problema de ler múltiplas *tags* em um pequeno espaço.

3.3 Triangulação

A triangulação é uma técnica de localização, que se baseia em propriedades geométricas e fórmulas matemáticas para a definição de pontos no espaço. Existem duas técnicas distintas de triangulação: a lateração e angulação. A distinção entre as duas é que para a lateração se utiliza de distâncias conhecidas entre um referencial e o ponto a ser encontrado, e na angulação, utilizamos ângulos formados pelo referencial e o ponto.

Através da técnica de lateração, obtemos a posição relativa da *tag* utilizando a distância entre ela e os Leitores dispostos no ambiente (que possuem posições fixas e cuja localização é conhecida). Para calcular a posição de um ponto num sistema de coordenadas tridimensional, é necessário conhecer as distâncias entre pelo menos quatro pontos não coplanares [25].

Como se obtêm apenas o tempo de sinal do retorno e o número de identificação da *tag*, foi utilizado o método de tempo de chegada, que consiste no cálculo da distância entre dois pontos através do tempo que um sinal demora a chegar de um ponto ao outro [25]. Na figura 3.6 temos uma visão de topo da técnica de lateração em um ambiente tridimensional.

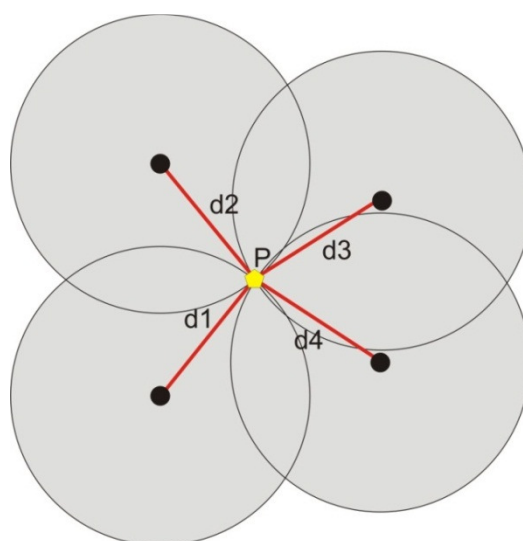


Figura 3.6 – Visão de topo da lateração tridimensional.

3.4 Sobre a Simulação nesse Trabalho

Este trabalho foi desenvolvido com base na simulação de Leitores/Receptores, do tipo *RFID Middle Range Integrated Reader* modelo SAAT-I801 [29], mostrado na figura 3.7.



Figura 3.7 – *RFID Middle Range Integrated Reader* SAAT-I801 [29].

Este aparelho tem por característica a utilização das frequências de 902-928 MHz e 865 – 868 MHz. Utiliza o protocolo ISO18000-6B para comunicação com as *tags*, com alcance de leitura de 0 a 8 metros e de escrita de 0 a 3 metros de distância entre a *tag* monitorada e o leitor/receptor. Este Leitor/Receptor apresenta um custo de U\$ 320,00 conforme apresentamos no apêndice C.

As *tags* utilizadas obedecem ao padrão EPC-96, e são *tags* passivas como mostra a figura 3.8, visando minimizar os custos. Seu custo varia de U\$ 0,15 a U\$ 1,00 [30].

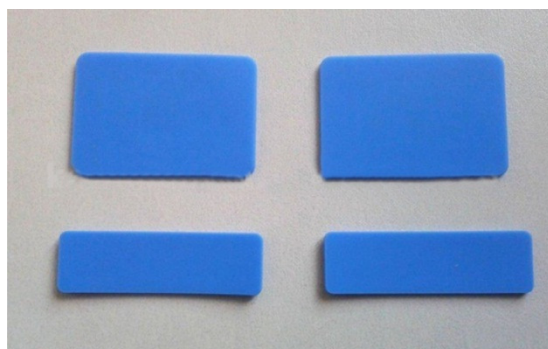


Figura 3.8 – Exemplo de *tag* utilizada neste sistema [30].

Capítulo 4

O Sistema Proposto

4.1 Visão do sistema SACCOB

O sistema SACCOB (Sistema de Alerta Contra Colisão com Obstáculos) foi desenvolvido para proporcionar às pessoas com deficiência uma maior segurança em relação a sua movimentação em espaços que contenham obstáculos. A ideia do sistema é, através de monitoramento, minimizar as colisões do indivíduo com os obstáculos presentes no ambiente em que ele se encontra. A partir de sensores dispostos em cada obstáculo e também no próprio indivíduo, o sistema é capaz de calcular a distância entre o obstáculo e o deficiente visual, avisando-o sobre a sua proximidade.

4.2 Modelagem

Foi utilizada a metodologia ágil XP [31] (*Extreme Programming*), pelo fato de ser um processo de desenvolvimento de software voltado, entre outras características, para:

- Projetos cujos requisitos são vagos e mudam com frequência;
- Desenvolvimento de sistemas Orientados a Objetos;
- Desenvolvimento iterativo ou incremental, onde o sistema começa a ser desenvolvido e ao longo do processo vai ganhando novas funcionalidades aumentando seu valor para o cliente.

A outra parte do desenvolvimento conta com a metodologia de desenvolvimento ágil SCRUM [32]. Decidimos por esta metodologia por ser um processo de desenvolvimento de software voltado, entre outras características, para:

- Cada *Sprint* (o desenvolvimento do sistema é dividido em ciclos regulares ao longo do tempo, e chamamos de *Sprint* cada um destes ciclos) é uma iteração que segue um ciclo e entrega incremento de software pronto.
- Há entrega de conjunto fixo de itens do *Backlog* (lista que contem as funcionalidades desejadas no sistema) em série de interações curtas ou *Sprints*;
- Breve sessão de planejamento, na qual os itens do *Backlog* para uma *Sprint* (iteração) são definidos;
- Retrospectiva, na qual todos os membros da equipe refletem sobre a *Sprint* passada.

Aliado a estas metodologias de desenvolvimento, será utilizado o padrão de arquitetura MVC (Modelo, Visão e Controlador) [33], onde o sistema será dividido em três camadas: o modelo, a visão e o controlador.

- Modelo: oferece a principal funcionalidade da aplicação, sendo ciente da dependência dos componentes de controle e visão;
- Visão: cada visão corresponde a um estilo e formato de apresentação de informações para o usuário;
- Controlador: utiliza as entradas dadas pelo usuário, a partir de eventos, e executa operações em cima do modelo.

4.2.1 Requisitos Funcionais

Os requisitos do sistema, que descrevem os serviços que o sistema deve oferecer, encontram-se listados de forma numerada. Nos Requisitos abaixo, o termo gerenciar, foi utilizado no sentido de inserir, alterar, excluir e consultar, essa simplificação foi feita para diminuir o volume de informações nesta seção do documento.

[RF – 01] Gerenciar Objetos

O sistema deverá permitir ao administrador gerenciar os objetos presentes no ambiente controlado, contendo alguns dados métricos (largura e comprimento). Disponibiliza funções de incluir, excluir, consultar e alterar os mesmos. Prioridade: Alta. Solicitante: Administrador e usuário.

[RF – 02] Gerenciar Pessoas

O sistema deverá permitir ao administrador gerenciar as pessoas que se utilizem do ambiente controlado, contendo alguns dados pessoais (nome, raio). Disponibiliza funções de incluir, excluir, consultar e alterar os mesmos. Prioridade: Alta. Solicitante: Administrador.

[RF – 03] Gerenciar Tag

O sistema deverá permitir ao administrador gerenciar os sensores que se vinculam a pessoas ou objetos, contendo alguns dados de referência (identificação e estado da *tag*). Disponibiliza funções de incluir, excluir, consultar e alterar os mesmos. Prioridade: Alta. Solicitante: Administrador.

[RF – 04] Gerenciar Leitor

O sistema deverá permitir ao administrador gerenciar os sensores que se vinculam a pessoas ou objetos, contendo alguns dados de referência (posição em relação aos eixos x, y, z). Disponibiliza funções de incluir, excluir, consultar e alterar os mesmos. Prioridade: Alta. Solicitante: Administrador.

[RF – 05] Gerenciar Local

O sistema deverá permitir ao administrador gerenciar os locais que se vinculam a sensores, poderemos validar o local de cada sensor se for necessário. Conterá dados de sua localidade (posições “x” e “y” em relação ao ambiente e seus tamanhos, comprimento e largura). Disponibiliza funções de incluir, excluir, consultar e alterar os mesmos. Prioridade: Alta. Solicitante: Administrador.

4.2.2 Requisitos Não Funcionais

O sistema elaborado, embora seja de cunho acadêmico, apresenta uma quantidade considerável de requisitos não funcionais para se obter o mínimo de qualidade requerida. Apresentamos estes requisitos, que são definidos por três tipos: requisitos de processo, requisitos de produto e requisitos externos.

Requisitos de Processo

[RNF/PROC-01] A modelagem do sistema deverá ser orientada a objetos.

[RNF/PROC-02] Preferencialmente deve ser usada uma metodologia ágil para o desenvolvimento do sistema, pois o mesmo não tem todos os requisitos definidos e novas funcionalidades poderão ser adicionadas.

Requisitos de Produto

[RNF/PER-03] Para obter eficiência na consulta dos dados armazenados deverá ser utilizado o banco de dados PostgreSQL com modelagem normalizada.

[RNF/PER-04] O sistema deverá simular o seu funcionamento em tempo real.

[RNF/CUS-05] O sistema será implementado em Java para a facilidade de portabilidade caso necessário.

[RNF/USA-06] Deve apresentar mensagens de erro quando o usuário tentar entrar com dados inconsistentes.

[RNF/USA-07] Para uma aparência agradável do sistema, utilizaremos tons de cores mais suaves, e padronização de telas.

[RNF/USA-08] O sistema implementa a arquitetura de software MVC, onde os modelos de dados, a parte visual e parte lógica de funcionamento são separadas, tornando mais fácil qualquer incremento no software.

Requisitos Externos

[RNF/EXT-09] O sistema deverá atender ao auxílio mínimo para uma pessoa com deficiência visual, sem influenciar sua privacidade.

4.2.3 Diagrama de Classes

A Figura 4.1 é apresentado o diagrama de Classes do sistema juntamente com uma breve discussão sobre o mesmo.

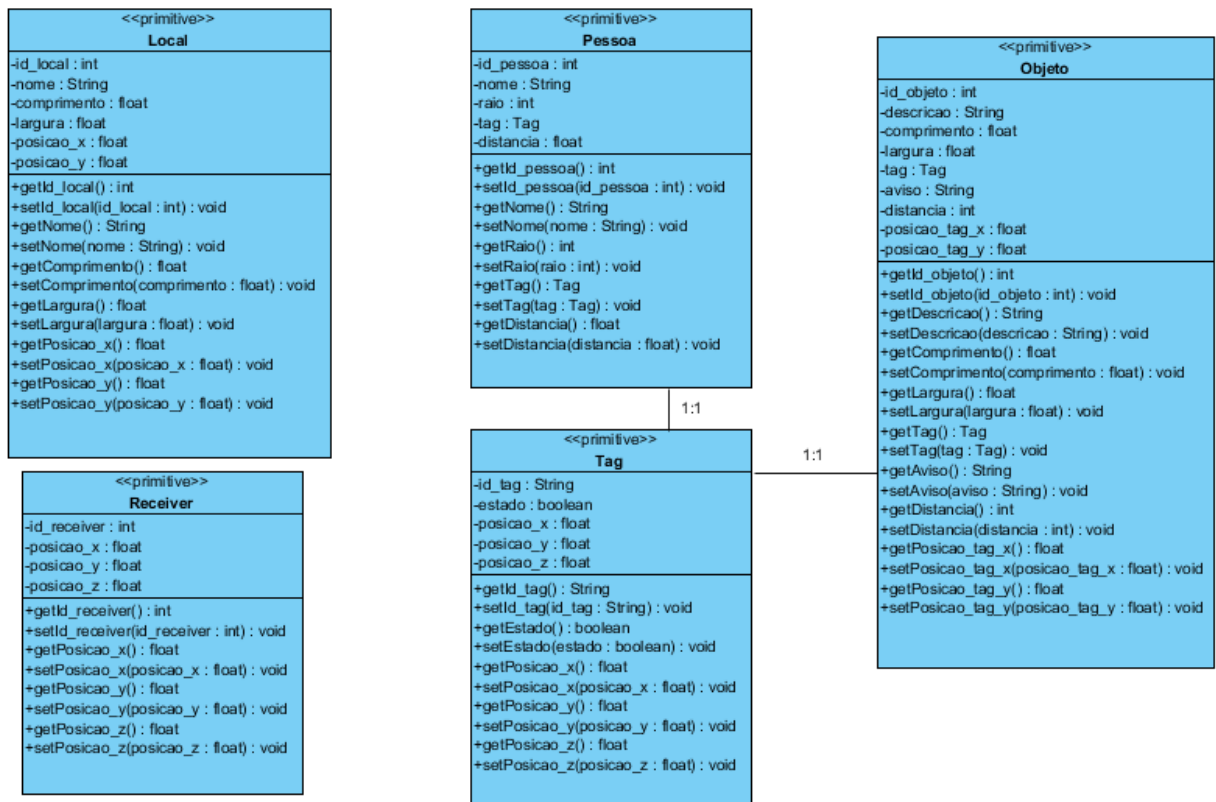


Figura 4.1 – Diagrama de Classes do sistema SACCQB

A classe Tag apresenta os atributos: *id_tag*, *estado*, *posicao_x*, *posicao_y*, *posicao_z*. Apresenta os métodos sets e gets para alterar e recuperar todos os atributos respectivamente. O atributo *estado* é utilizado para saber se a tag está associada a algum objeto ou pessoa, e os atributos *posição_x*, *posição_y* e *posição_z*, são utilizados para achar a posição exata da pessoa ou objeto vinculado.

A classe Pessoa apresenta os atributos: *id_pessoa*, *nome*, *raio*, *distância* e *tag*. Apresenta os métodos sets e gets para alterar e recuperar todos os atributos respectivamente. O atributo *raio* simboliza o corpo de uma pessoa. O atributo *distância* é utilizado para verificar a distância entre a pessoas, onde é pego o valor do atributo distância da pessoa presente no ambiente e é realizado a distância entre dois pontos entre esta pessoa e a pessoa que está sendo mapeada, onde esta distância tem de ser maior que a soma dos raios de ambas as pessoas e o atributo distância da pessoa presente no ambiente para que não seja disparado o aviso para a pessoa mapeada. O atributo *tag*, é a tag que está vinculada a esta pessoa.

A classe Objeto apresenta os atributos: *id_objeto*, *descricao*, *comprimento*, *largura*, *tag*, *aviso*, *distância*, *posicao_tag_x* e *posicao_tag_y*. Apresenta os métodos sets e gets para alterar

e recuperar todos os atributos respectivamente. Os atributos *comprimento* e *largura* representam as dimensões do objeto em um espaço bidimensional. O atributo *distância* é utilizado para verificar a distância entre a pessoa mapeada e o objeto. O atributo *tag*, é a tag que está vinculada a este objeto. Os atributos *posicao_tag_x* e *posicao_tag_y* são referentes à posição em que a *tag* se encontra fixada no objeto, a partir do topo esquerdo do objeto.

A classe Local apresenta os atributos: *id_local*, *nome*, *comprimento*, *largura*, *posicao_x* e *posicao_y*. Apresenta os métodos sets e gets para alterar e recuperar todos os atributos respectivamente. Os atributos *comprimento* e *largura* representam as dimensões do local em um espaço bidimensional. Os atributos *posicao_x* e *posicao_y* constituem a posição inicial do local. Utilizamos a classe Local para filtrar as *tags* para que o Módulo de Teste de Colisão verifique apenas as *tags* que se encontram no mesmo local que a pessoa mapeada.

A classe Receiver apresenta os atributos: *id_receiver*, *posicao_x*, *posicao_y* e *posicao_z*. Apresenta os métodos sets e gets para alterar e recuperar todos os atributos respectivamente. Os atributos *posicao_x*, *posicao_y* e *posicao_z* constituem a posição física de cada Leitor/Receptor, que é utilizada para o mapeamento de cada uma das *tags*.

4.3 Síntese de Voz

Para sintetização de voz, foi utilizada a biblioteca Java Speech [34], onde por sua vez, podem ser configurados alguns parâmetros para a voz sintetizada. Na figura 4.2 é apresentado trecho de código onde a sintetização é realizada.

```
String texto = " <JSML> " + "<BREAK MSECS=\"0\"/>" + "<PROS PITCH=\"98\" RATE=\"200\">"
              + objeto.get(0).getAviso()
              + "</PROS>" + " </JSML> ";
Sintetizador.speak(texto);
```

Figura 4.2 – Exemplo da Sintetização de Voz.

A partir da figura 4.2, pode-se visualizar os parâmetros “BREAK MSECS”, que se refere ao tempo de espera antes de iniciar a sintetização, o “PROS PITCH”, referente à tonalidade da voz sintetizada e “RATE” que se refere à velocidade da sintetização.

4.4 Base de dados

Utilizamos o SGBD PostgreSQL [35] para a armazenagem e vínculos dos sensores à objetos e pessoas. Foi utilizado o padrão DAO (*Data Access Object*) [36] para o acesso ao SGBD junto ao padrão *Factory Method* [37], para que assim o sistema tenha uma boa divisão para acompanhar a metodologia de desenvolvimento.

4.4.1 Diagrama Entidade Relacional da Base de Dados

Para uma discussão mais apurada sobre a utilização da base de dados, é apresentada a figura 4.3 referente ao diagrama de entidade relacional da Base de Dados do sistema SACCOB.

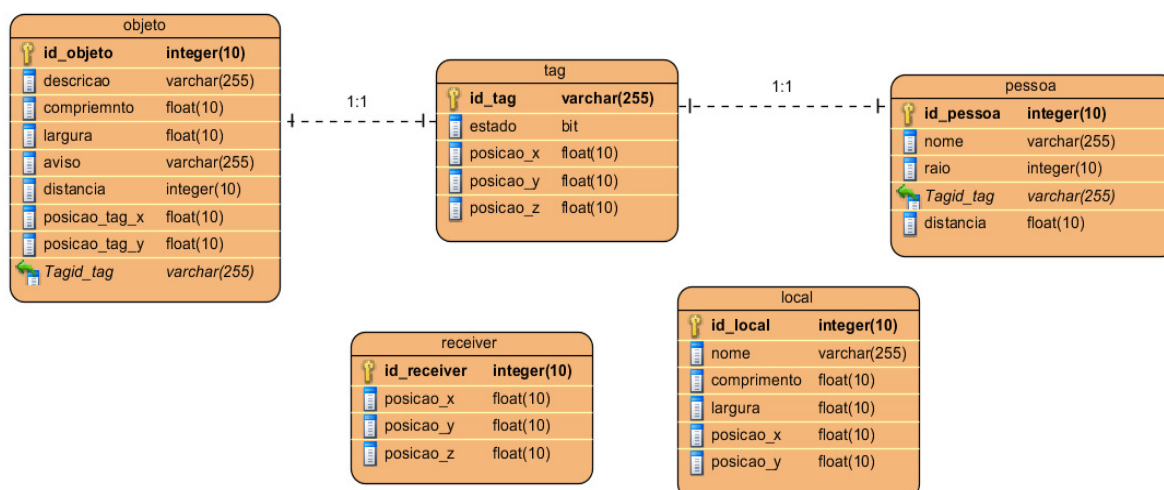


Figura 4.3 – Diagrama Entidade Relacional da Base de Dados do sistema SACCOB

A decisão da utilização da base de dados foi devido à necessidade de saber a que objeto ou pessoa a tag esteja representando, sem a necessidade de que esta tag seja ativa, podendo assim utilizar tags passivas para diminuir custos para este trabalho. A ideia de armazenamento dos locais é para que os testes de colisão sejam feitos apenas com tags que estejam no mesmo local da pessoa mapeada. É preciso ter as posições dos quatro Leitores/Receptores, para que os mapeamentos (triangulação) possam ser realizados.

4.5 Módulos de Leitura de *Tags* e Teste de Colisão

O Módulo de Leitura de *Tags* e o Módulo de Teste de Colisão foram unidos e desenvolvidos com embasamento em programação concorrente, onde os dois se tornam um *thread* que segundo Silbestschatz [38], são chamados de processos leves. Estes processos considerados unidades básicas de utilização da CPU, onde cada um do mesmo possui um ID, um contador de programa, um conjunto de registradores e uma pilha, para implementar os módulos de evento. Para tanto, teremos um sistema que possui múltiplos *threads* conseguindo assim, realizar mais de uma tarefa por vez.

A ideia de utilização de programação concorrente nestes módulos vem com o pensamento de poder mapear mais de uma pessoa ao mesmo tempo, porém não fará parte desse trabalho. Pensando nisto, já existe um suporte primário para que esta nova fase do sistema possa ser realizada, para que não ocorra inconsistência de dados (ler um dado que não está atualizado), utilizando um recurso de sincronização do Java, chamado Synchronized [38].

4.5.1 Módulo de Leitura de *Tags*

Neste módulo, a partir de técnica de lateração vinculada ao método de tempo de chegada, é realizado o cálculo da distância entre dois pontos (*tag* e Leitor). Como o raio de leitura do Leitor/Receptor é esférico trabalhando em um ambiente tridimensional, é necessário que a triangulação realizada seja tridimensional. Para tal, são utilizados quatro Leitores não coplanares. São realizadas as leituras das *tags* ativando um Leitor por vez, para que não ocorram leituras simultâneas, e estes valores são armazenados em uma lista que armazena o identificador da *tag* e o tempo de resposta da mesma. Após a conclusão da fase de armazenagem, é realizada a operação de triangulação resolvendo o sistema linear apresentado na Equação 4.1.

$$\begin{cases} 2x(x_2 - x_1) + 2y(y_2 - y_1) + 2z(z_2 - z_1) = d_1^2 - d_2^2 - x_1^2 + x_2^2 - y_1^2 + y_2^2 - z_1^2 + z_2^2 \\ 2x(x_3 - x_2) + 2y(y_3 - y_2) + 2z(z_3 - z_2) = d_2^2 - d_3^2 - x_2^2 + x_3^2 - y_2^2 + y_3^2 - z_2^2 + z_3^2 \\ 2x(x_4 - x_3) + 2y(y_4 - y_3) + 2z(z_4 - z_3) = d_3^2 - d_4^2 - x_3^2 + x_4^2 - y_3^2 + y_4^2 - z_3^2 + z_4^2 \end{cases} \quad (4.1)$$

Onde:

x , y e z correspondem à coordenada da *tag* mapeada.

x_1 , y_1 e z_1 correspondem à coordenada do primeiro Leitor/Receptor.

x_2, y_2 e z_2 correspondem à coordenada do segundo Leitor/Receptor.

x_3, y_3 e z_3 correspondem à coordenada do terceiro Leitor/Receptor.

x_4, y_4 e z_4 correspondem à coordenada do quarto Leitor/Receptor.

d_1 corresponde à distância entre a *tag* e o primeiro Leitor/Receptor.

d_2 corresponde à distância entre a *tag* e o segundo Leitor/Receptor.

d_3 corresponde à distância entre a *tag* e o terceiro Leitor/Receptor.

d_4 corresponde à distância entre a *tag* e o quarto Leitor/Receptor.

Para encontrar a distância, foi utilizada a Equação 4.2 [25].

$$d = v.T \quad (4.2)$$

Onde:

d corresponde à distância entre a *tag* e o leitor/receptor.

v corresponde à velocidade da onda de rádio frequência.

T corresponde ao tempo de resposta da *tag*.

Para encontrar a velocidade da onda de rádio frequência, foi utilizada a Equação 4.3 [39].

$$v = \lambda \cdot f \quad (4.3)$$

Onde:

v corresponde à velocidade da onda de rádio frequência.

λ corresponde ao comprimento de onda (corresponde a um valor de).

f corresponde à frequência da onda.

Foram considerados os valores de $\lambda = 0,3333$ metros [40] e $f = 900.10^6$ hertz (frequência em que o leitor/receptor está funcionando). Resolvendo a Equação 4.3, obtêm-se $v = 299,97.10^6 m / s$.

Com as coordenadas dos quatro leitores, e a distância entre os leitores e a *tag* são conhecidas, utilizou-se o método Eliminação de Gauss [41] para resolver este sistema linear,

encontrando assim a posição real da *tag* como é apresentado no Apêndice A, onde encontramos o código fonte do Módulo de Leitura de *Tags*.

4.5.2 Módulo de Teste de Colisão

Neste módulo, já se sabem as posições exatas de cada uma das *tags* (posição no eixo x, no eixo y e no eixo z, porém nesta etapa descartamos o valor do eixo z, pois nossos cálculos passam a ser bidimensionais), sendo assim, o sistema com a relação *tag*/pessoa pode identificar a *tag* referente à pessoa mapeada. A partir deste momento, a pessoa pode estar em nove posições, como mostra a figura 4.4, em relação a um objeto. Para a realização do cálculo de distância no teste de colisão, é necessário que se utilize o ponto do objeto mais próximo à pessoa.

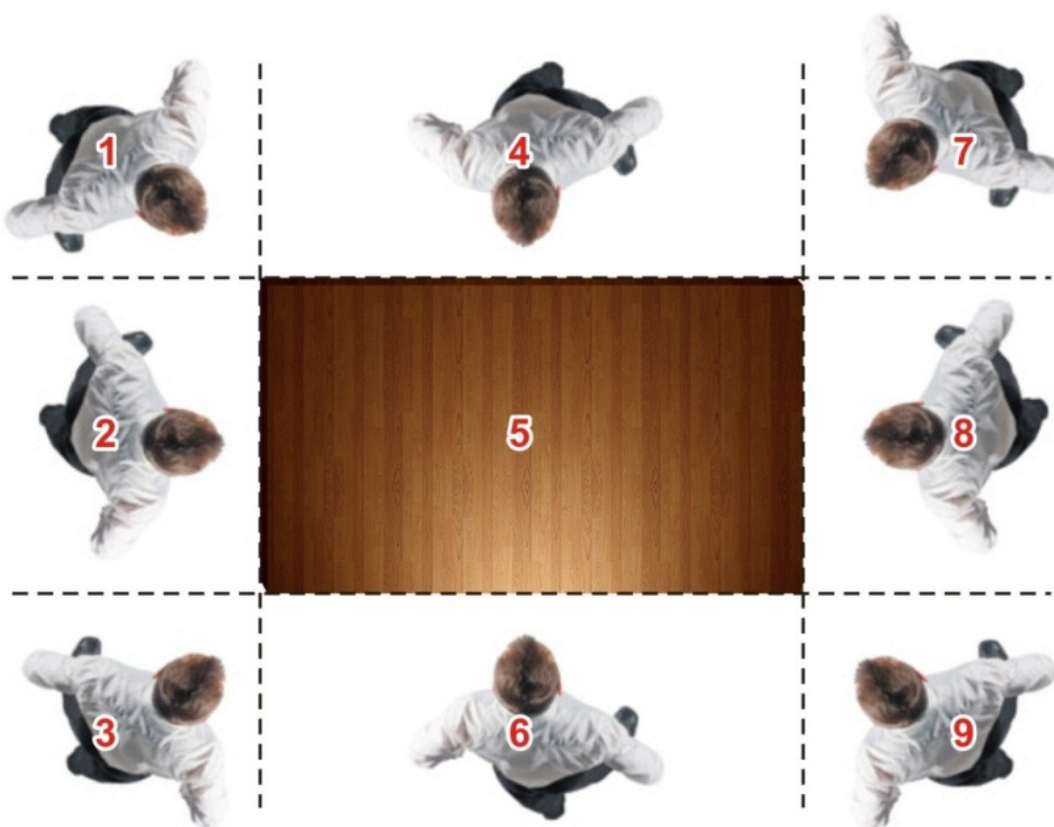


Figura 4.4 – Locais em que a pessoa pode se encontrar em relação a um objeto.

Depois de selecionado os pontos do objeto, e se a posição da pessoa em relação ao objeto não for a posição cinco da figura 4.4 - pois neste trabalho não é considerado colisão se a

pessoa mapeada estiver em cima do objeto - é realizado o método de distância entre dois pontos. Se esta distância for menor que a distância que o usuário deve manter do objeto mais o raio da mesma, chama-se o sintetizador de voz, que por sua vez sintetiza o texto de aviso daquele objeto, avisando a pessoa mapeada. No apêndice B encontra-se o código fonte do teste de colisão.

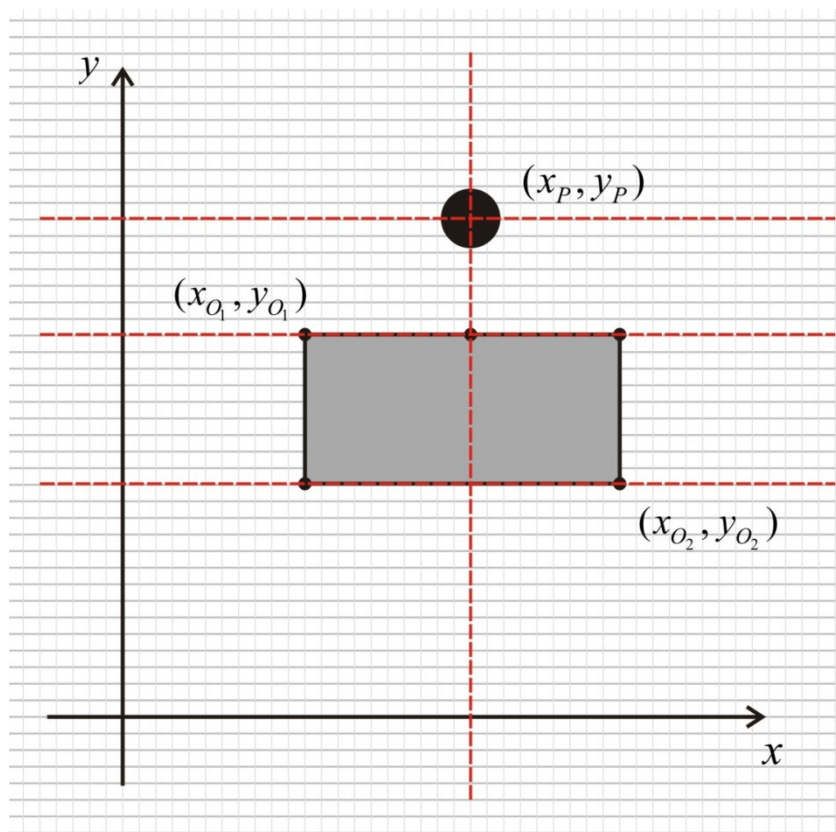


Figura 4.5 – Teste de colisão a partir do plano cartesiano.

Com a figura 4.5 é possível compreender como é encontrado o ponto do objeto mais próximo à pessoa mapeada.

4.6 Funcionamento

Na Figura 4.6 é apresentado o diagrama de Fluxo do sistema, onde é possível fazer a verificação da sua estrutura, identificando sua divisão em seis fases e cada qual responsável por uma função.

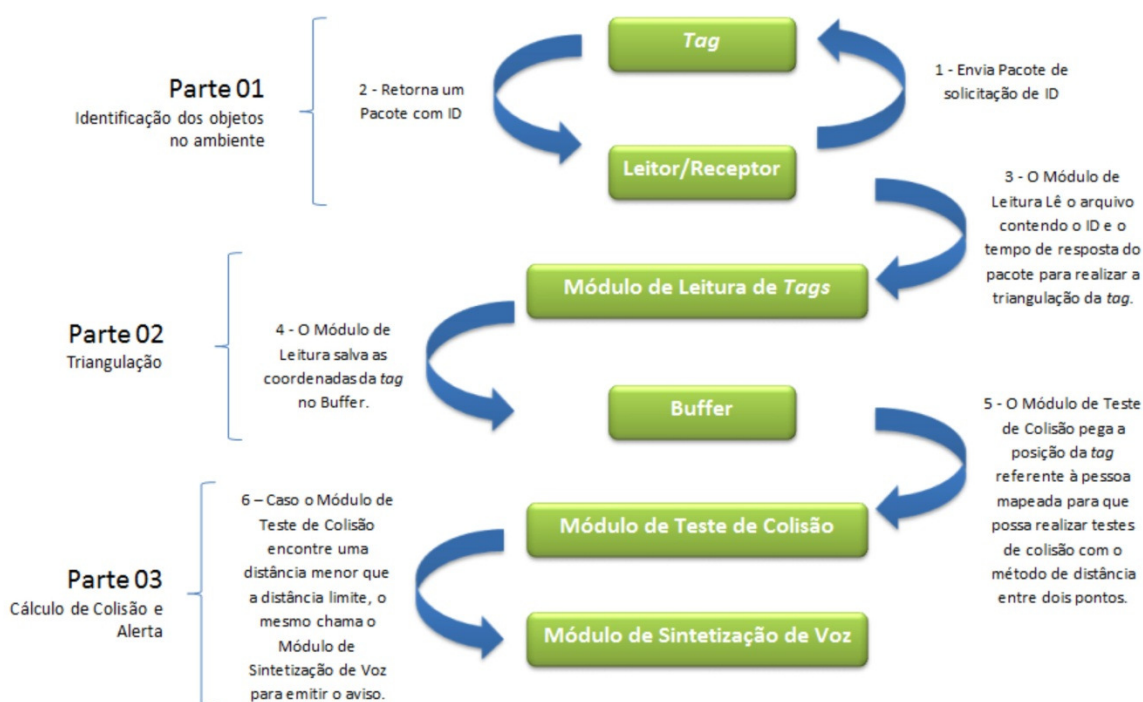


Figura 4.6 – Diagrama Fluxo do sistema SACCOB.

O módulo Buffer trata-se de uma lista estática que armazena as coordenadas cartesianas “x”, “y” e “z” de cada uma das *tags*.

A fase 1 e a fase 2 não são implementadas nesse trabalho, visto que se torna necessário possuir o SDK (*Software Developers Kit* – pacote de desenvolvimento de software) do Leitor/Receptor RFID. Para a fase 3, o Módulo de Leitura de *Tags* realiza leituras de arquivos de texto, que contém o identificador da *tag* e o tempo de resposta, provenientes dos Leitores/Receptores. Na fase 4, o Módulo de Leitura de *Tags* efetua os cálculos de triangulação e armazena todas as posições das *tags* no Buffer.

Já na fase 5, o módulo de Teste de Colisão solicita a posição da pessoa mapeada ao Buffer, com isto é possível calcular em que local a pessoa mapeada se encontra. A partir deste momento são lidas as posições de cada uma das *tags*, com exceção da *tag* da pessoa mapeada, validando o local e calculando a distância entre a pessoa mapeada e o obstáculo, como apresentado na seção 4.5.2. Se alguma destas distâncias for menor que a distância limite, o Módulo de Teste de Colisão faz uma chamada para o Módulo de Sintetização de Voz com o aviso a ser sintetizado que acontece na fase 6. No apêndice D são apresentadas as telas de interface do sistema, com uma breve explicação sobre seu funcionamento.

4.7 Análises dos Resultados

Nesta seção são apresentados e discutidos os resultados obtidos no desenvolvimento deste trabalho.

Foram criadas cenas com a rota de todos os objetos e pessoas previamente determinados, e analisados os índices de formação de fila nos avisos emitidos pelo sistema, contudo, as separações dos locais dentro do ambiente foram feitas através de divisórias e não por paredes, pelo fato da interferência ou falha de sinal de radiofrequência ao atravessar as paredes. Destas cenas analisadas, tendo uma média de quinze possíveis colisões (emissão de avisos), o sistema atingiu sua funcionalidade, com um índice de emissão de avisos, sem formação de filas, de onze a treze avisos, apresentou um fator médio de minimização de colisão de 86%. As colisões que não foram evitadas são geradas devido ao Módulo de Sintetização enfileirar os avisos um a um, e apenas sintetizar o aviso quando o aviso que se encontra primeiro na fila for concluído.

Ao analisar o gráfico que contém o tempo de execução do sistema, representado pela figura 4.7, obteve-se uma média de 12 milissegundos de execução. Para se atingir este tempo de execução foi utilizado um notebook com processador intel core i-5, com 6 GB de memória ram, HD de 500 GB, e placa de vídeo ATI-Radeon HD 6630 de 1GB.

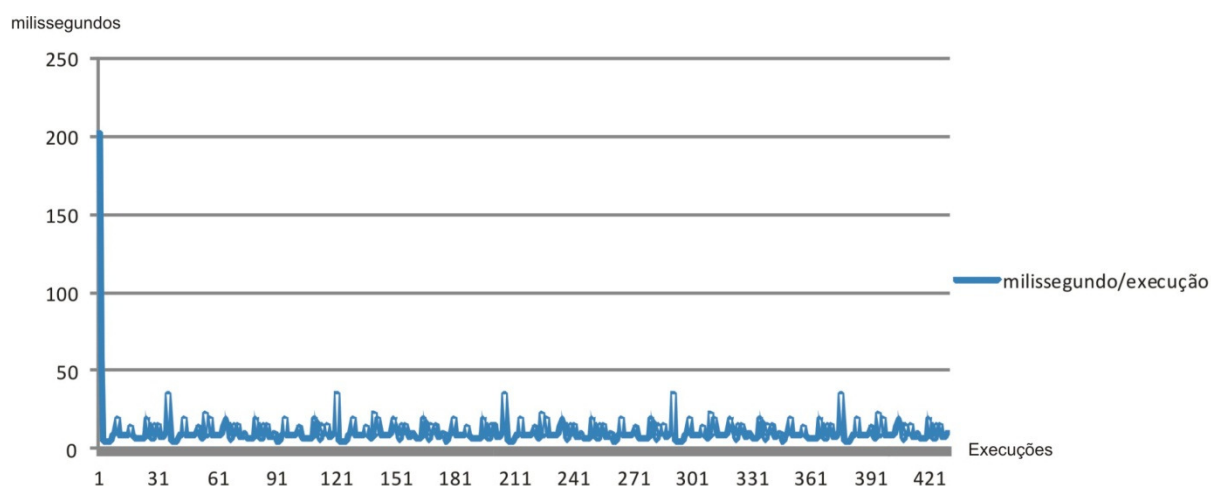


Figura 4.7 – Gráfico de tempo de Execução do sistema SACCOB.

Capítulo 5

Conclusões

5.1 Considerações finais

A proposta para esse trabalho é a minimização do índice de colisão de uma pessoa com deficiência visual. Em relação aos custos de investimento para o sistema, são considerados valores intermediários, porém não de baixo custo, se for levar em conta todo o material necessário para o seu devido funcionamento. Por tanto, para que se efetive o desenvolvimento prático é preciso adquirir algumas ferramentas, sendo elas: um servidor (computador que irá executar o sistema), quatro leitores/escritores RFID, uma *tag* para cada pessoa e objeto, e um fone de ouvido *bluetooth*. É disposto na tabela 5.1, a relação dos custos estimados para equipamento.

Tabela 5.1 – custo estimado dos equipamentos utilizados no desenvolvimento do sistema

Qnt.	Ferramentas	Valor Unitário	Valor Total
01	Servidor do sistema	R\$ 1.200,00	R\$ 1.200,00
04	Leitores/Escritores RFID	R\$ 1200,00	R\$ 4.800,00
14	<i>Tags</i> Passivas	R\$ 10,00	R\$ 140,00
01	Fone de Ouvido <i>Bluetooth</i>	R\$ 150,00	R\$ 150,00

Para a utilização deste sistema em um ambiente com paredes, é necessário que se tenha um conjunto de 04 Leitores/Escritores RFID por cômodo, para que o sistema funcione.

A utilização do sistema RFID foi de grande importância para esse trabalho, pelo fato de possibilitar o encontro de posições muito próximas em que a pessoa mapeada e os obstáculos se encontram no ambiente, auxiliando no funcionamento do módulo de teste de colisão, que utiliza tanto a posição da pessoa mapeada quanto a dos obstáculos. Portanto, fornece uma locomoção mais segura para a pessoa com deficiência visual, que estará fazendo o uso deste sistema.

Os resultados apresentados de acordo com os objetivos do sistema podem dizer que de maneira positiva, o sistema mostrou uma função coerente em relação aos avisos emitidos para impedir colisões,

levando em conta que este trabalho reconhece apenas objetos com dimensões retangulares, e pessoas como circunferências. Mesmo apresentando esta característica e considerando que uma pessoa com deficiência visual, assim que escute o aviso sintetizado pelo sistema, fique atenta e reduza a sua velocidade, por frações de segundo, para evitar as colisões.

Conforme análises realizadas no decorrer do desenvolvimento do sistema, percebemos que com o afinamento da configuração do sistema como um todo, se obtém um índice de até 86% na redução das colisões. Temos que considerar o índice de 14% como alguns atrasos ou formação de filas para a síntese dos avisos que podem ocorrer durante o funcionamento e o uso incorreto do mesmo.

Obteve-se resultados de tempo de execução satisfatórios com uma média de 12 milissegundos para o funcionamento do Módulo de Leitura de *Tags*, juntamente com o Módulo de Teste de Colisão como apresentado na sessão 4.7. Porém, é possível que em alguns momentos, essa média de tempo de execução obtido não seja o suficiente para atender às necessidades de uma determinada ação que ocorra em um espaço de tempo menor, apresentando então, uma limitação para este sistema.

5.2 Perspectivas futuras

Muito pode ser feito com base neste trabalho. Uma evolução importante para este sistema seria o reconhecimento de objetos com padrões diferentes de padrões retangulares, tornando possível o mapeamento de qualquer forma dentro do espaço mapeado seja ele um cilindro, tal como um cone. Ainda nesta mesma linha de raciocínio, este sistema se limita apenas a translações no objeto, mas não suporta operações de rotação sobre o mesmo. Ao se aplicar uma rotação em um objeto, apenas tendo uma *tag* associada a ele, não é possível o mapeamento de forma consistente, sendo necessário pelo menos duas *tags* para a realização de rotação, como mostrado na figura 5.1. Com estas duas evoluções o sistema se tornará muito mais preciso em seus cálculos.

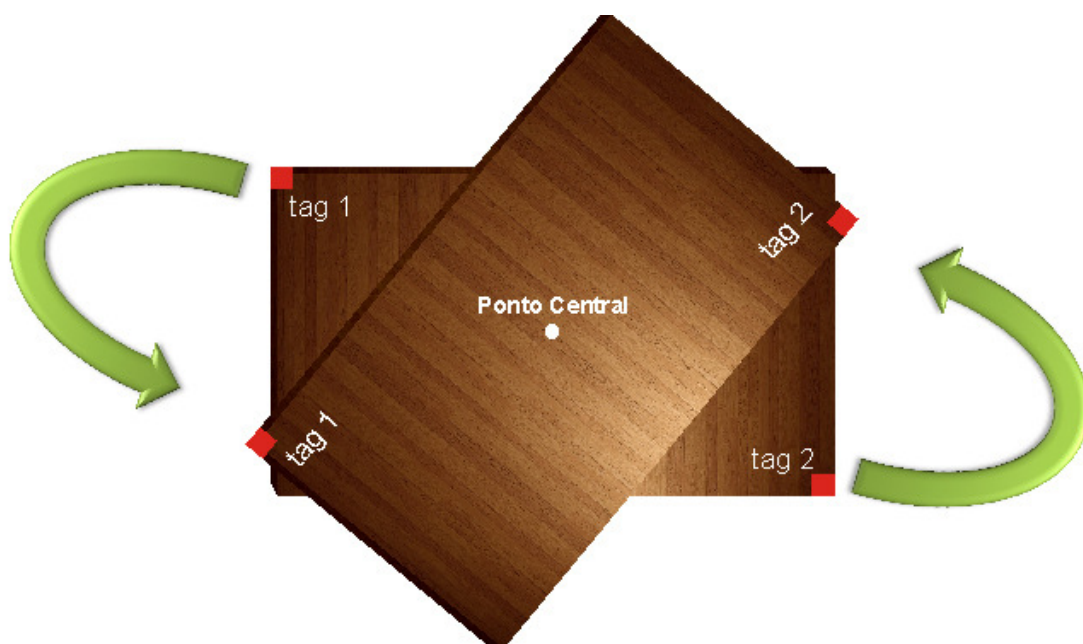


Figura 5.1 – Possibilidade de Calcular a rotação de um objeto para este trabalho.

Não nos prendendo apenas aos objetos, uma evolução muito significativa seria a implementação de toda a interface de configuração para que interaja com usuários de baixa visão, onde o mesmo possa realizar as configurações do sistema sem que o administrador do mesmo precise realiza-lo.

Em relação a tempo de execução, o tempo de 12 milissegundos pode não ser um tempo hábil considerado como tempo real. Uma possível otimização dos códigos fonte pode levar a resultados mais satisfatórios.

O dispositivo de áudio utilizado neste trabalho é limitado em relação à distância. A construção de um dispositivo de áudio móvel, que acompanhe o deficiente visual, para substituir o fone de ouvido *bluetooth 3.0*. Uma troca de tecnologia se torna viável. Caso este novo dispositivo fosse desenvolvido utilizando a rede wireless 802.11, teria um alcance muito maior.

Apêndice A

Código fonte do Módulo de Leitura das tags

O Módulo de leitura das *Tags* é dividido em dois métodos: o `lerDados` que é responsável por ler os dados dos arquivos texto, e o `lerTags`, que é responsável por calcular a posição de cada uma das tags.

```
public void leitorTag() {
    int deslocamento = Buffer.buffer.size();
    float x;
    float y;
    float z;
    float m;
    float d1;
    float d2;
    float d3;
    float d4;
    float matriz[][] = new float[3][4];
    ControllerReceiver controllerReceiver = new ControllerReceiver();
    List<Receiver> receiver = controllerReceiver.listar();
    if (incremento < receptor1.size()) {
        for (int k = 0; k < deslocamento; k++) {
            //joga os valores para a matriz
            d1 = receptor1.get(incremento).getTempo_retorno() * 299970000;
            d2 = receptor2.get(incremento).getTempo_retorno() * 299970000;
            d3 = receptor3.get(incremento).getTempo_retorno() * 299970000;
            d4 = receptor4.get(incremento).getTempo_retorno() * 299970000;

            matriz[0][0] = 2 * (receiver.get(1).getPosicao_x() - receiver.get(0).getPosicao_x());
            matriz[0][1] = 2 * (receiver.get(1).getPosicao_y() - receiver.get(0).getPosicao_y());
            matriz[0][2] = 2 * (receiver.get(1).getPosicao_z() - receiver.get(0).getPosicao_z());
            matriz[0][3] = (float) (Math.pow(d1, 2) - Math.pow(d2, 2)
                - Math.pow(receiver.get(0).getPosicao_x(), 2)
                + Math.pow(receiver.get(1).getPosicao_x(), 2)
                - Math.pow(receiver.get(0).getPosicao_y(), 2)
                + Math.pow(receiver.get(1).getPosicao_y(), 2)
                - Math.pow(receiver.get(0).getPosicao_z(), 2)
                + Math.pow(receiver.get(1).getPosicao_z(), 2));
            matriz[1][0] = 2 * (receiver.get(2).getPosicao_x() - receiver.get(1).getPosicao_x());
```

```

matriz[1][1] = 2 * (receiver.get(2).getPosicao_y() - receiver.get(1).getPosicao_y());
matriz[1][2] = 2 * (receiver.get(2).getPosicao_z() - receiver.get(1).getPosicao_z());
matriz[1][3] = (float) (Math.pow(d2, 2) - Math.pow(d3, 2)
- Math.pow(receiver.get(1).getPosicao_x(), 2)
+ Math.pow(receiver.get(2).getPosicao_x(), 2)
- Math.pow(receiver.get(1).getPosicao_y(), 2)
+ Math.pow(receiver.get(2).getPosicao_y(), 2)
- Math.pow(receiver.get(1).getPosicao_z(), 2)
+ Math.pow(receiver.get(2).getPosicao_z(), 2));
matriz[2][0] = 2 * (receiver.get(3).getPosicao_x() - receiver.get(2).getPosicao_x());
matriz[2][1] = 2 * (receiver.get(3).getPosicao_y() - receiver.get(2).getPosicao_y());
matriz[2][2] = 2 * (receiver.get(3).getPosicao_z() - receiver.get(2).getPosicao_z());
matriz[2][3] = (float) (Math.pow(d3, 2) - Math.pow(d4, 2)
- Math.pow(receiver.get(2).getPosicao_x(), 2)
+ Math.pow(receiver.get(3).getPosicao_x(), 2)
- Math.pow(receiver.get(2).getPosicao_y(), 2)
+ Math.pow(receiver.get(3).getPosicao_y(), 2)
- Math.pow(receiver.get(2).getPosicao_z(), 2)
+ Math.pow(receiver.get(3).getPosicao_z(), 2));
//calcula o sistema
for (int i = 0; i < 2; i++) {
for (int w = i + 1; w < 3; w++) {
m = matriz[w][i] / matriz[i][i];
matriz[w][i] = 0;
for (int j = i + 1; j < 4; j++) {
matriz[w][j] -= m * matriz[i][j];
}
}
}
//calcula os valores x, y e z de cada tag
z = matriz[2][3] / matriz[2][2];
y = (matriz[1][3] - (matriz[1][2] * z)) / matriz[1][1];
x = (matriz[0][3] - (matriz[0][2] * z) - (matriz[0][1] * y)) / matriz[0][0];
//armazena no buffer a coordenada da tag
Buffer.atualizarSensor(receptor1.get(incremento).getId_tag(), x, y, z);
incremento++;
}
}
}

public void lerDados() {
try {
receptor1 = new ArrayList<>();
receptor2 = new ArrayList<>();
receptor3 = new ArrayList<>();
receptor4 = new ArrayList<>();
leitura = new ArrayList<>();
RandomAccessFile ran1 = new RandomAccessFile(new File("arquivo1.txt"), "rw");

```

```

RandomAccessFile ran2 = new RandomAccessFile(new File("arquivo2.txt"), "rw");
RandomAccessFile ran3 = new RandomAccessFile(new File("arquivo3.txt"), "rw");
RandomAccessFile ran4 = new RandomAccessFile(new File("arquivo4.txt"), "rw");
String s;
ArrayList<String[]> dadosEntrada1 = new ArrayList<>();
ArrayList<String[]> dadosEntrada2 = new ArrayList<>();
ArrayList<String[]> dadosEntrada3 = new ArrayList<>();
ArrayList<String[]> dadosEntrada4 = new ArrayList<>();

for (int i = 0; i < 4544; i++) {
    s = ran1.readLine();
    dadosEntrada1.add(s.split("/"));
    s = ran2.readLine();
    dadosEntrada2.add(s.split("/"));
    s = ran3.readLine();
    dadosEntrada3.add(s.split("/"));
    s = ran4.readLine();
    dadosEntrada4.add(s.split("/"));
}
for (int i = 0; i < dadosEntrada1.size(); i++) {
    receptor1.add(new Leitura(dadosEntrada1.get(i)[0],
    ValidaDados.tratarValor(dadosEntrada1.get(i)[1])));
    receptor2.add(new Leitura(dadosEntrada2.get(i)[0],
    ValidaDados.tratarValor(dadosEntrada2.get(i)[1])));
    receptor3.add(new Leitura(dadosEntrada3.get(i)[0],
    ValidaDados.tratarValor(dadosEntrada3.get(i)[1])));
    receptor4.add(new Leitura(dadosEntrada4.get(i)[0],
    ValidaDados.tratarValor(dadosEntrada4.get(i)[1])));
}
Posicao(ValidaDados.tratarValor(dadosEntrada.get(i)[1]),
} catch (Exception ex) {
    System.out.println(ex);
}
}
}

```

Apêndice B

Código fonte do Módulo de Teste de Colisão

```
public void testeColisao() {
    Sintetizador vp = new Sintetizador();
    //é pego a posição da pessoa mapeada
    Pessoa p = Principal.u;
    Posicao posicao = Buffer.lerTag(p.getTag().getId_tag());
    float x = posicao.getX();
    float y = posicao.getY();
    //faz o mapeamento do local em que a pessoa se encontra
    ControllerLocal controllerLocal = new ControllerLocal();
    List<Local> local = controllerLocal.listar();
    Local local_mapeado = new Local(0, null, 0, 0, 0, 0);
    ControllerPessoa controllerPessoa = new ControllerPessoa();
    //varre todo o buffer e verifica se alguma tag se encontra no local em que a pessoa
mapeada se encontra
    ControllerObjeto controllerObjeto = new ControllerObjeto();
    for (int i = 0; i < Buffer.buffer.size(); i++) {
        posicao = Buffer.lerTag(p.getTag().getId_tag());
        for (int j = 0; j < local.size(); j++) {
            float x_local = local.get(j).getPosicao_x() + local.get(j).getComprimento();
            float y_local = local.get(j).getPosicao_y() + local.get(j).getLargura();
            if ((x >= local.get(j).getPosicao_x()) && (x <= x_local)
                && (y >= local.get(j).getPosicao_y()) && (y <= y_local)) {
                local_mapeado = local.get(j);
                break;
            }
        }
    }
}
```

```

    }
}
if ((Buffer.buffer.get(i).getPosicao_x() >= local_mapeado.getPosicao_x())
    && (Buffer.buffer.get(i).getPosicao_x() <= (local_mapeado.getPosicao_x() +
local_mapeado.getComprimento()))
    && (Buffer.buffer.get(i).getPosicao_y() >= local_mapeado.getPosicao_y())
    && (Buffer.buffer.get(i).getPosicao_y() <= (local_mapeado.getPosicao_y() +
local_mapeado.getLargura()))
    && (Buffer.buffer.get(i).getPosicao_x() != posicao.getX())
    && (Buffer.buffer.get(i).getPosicao_y() != posicao.getY())) {
//verifico se a tag é referente a uma pessoa ou a um objeto
List<Pessoa> pessoa = controllerPessoa.listarPernsonalizado("Tag", "" +
Buffer.buffer.get(i).getId_tag());
if (!pessoa.isEmpty()) {
    Posicao pos = new Posicao(Buffer.buffer.get(i).getPosicao_x(),
    Buffer.buffer.get(i).getPosicao_y(),
    Buffer.buffer.get(i).getPosicao_z());
    float distância = (float) Math.sqrt((Math.pow((x - pos.getX()), 2) + Math.pow((y -
pos.getY()), 2)));
    if (distância < (pessoa.get(0).getDistância() + p.getRaio() + pessoa.get(0).getRaio())) {
        if (!validadeAviso[i]) {
            String texto = " <JSML> " + "<BREAK MSECs=\"0\"/>" + "<PROS PITCH=\"98\"
RATE=\"200\">"
            + "Você está próximo de " + pessoa.get(0).getNome()
            + "</PROS>" + "</JSML> ";
            Sintetizador.speak(texto);
            validadeAviso[i] = true;
        }
    } else {
        validadeAviso[i] = false;
    }
} else {

```

```

//verifica se a Tag é referente a um objeto e qual o mesmo
List<Objeto> objeto = controllerObjeto.listarPermnalizado("Tag",
Buffer.buffer.get(i).getId_tag());
if (!objeto.isEmpty()) {
    Posicao pos;
    //recebe posição inicial e final do objeto através de cálculos
    float xio = Buffer.buffer.get(i).getPosicao_x() - objeto.get(0).getPosicao_tag_x();
    float xfo = xio + objeto.get(0).getComprimento();
    float yio = Buffer.buffer.get(i).getPosicao_y() - objeto.get(0).getPosicao_tag_y();
    float yfo = yio + objeto.get(0).getLargura();
    //verifica a posição da pessoa mapeada em relação ao objeto
    if ((x < xio) && (y <= yio)) { // ok - posição 1
        pos = new Posicao(xio, yio, 0);
    } else {
        if ((x < xio) && ((y >= yio) && (y <= yfo))) { //ok - posição 2
            pos = new Posicao(xio, y, 0);
        } else {
            if ((x <= xio) && (y > yfo)) { //ok - posição 3
                pos = new Posicao(xio, yfo, 0);
            } else {
                if (((x >= xio) && (x <= xfo)) && (y < yio)) { //ok - posição 4
                    pos = new Posicao(x, yio, 0);
                } else {
                    if ((x >= xio) && (x <= xfo) && ((y >= yio) && (y <= yfo))) { //ok - posição 5
                        pos = new Posicao(x, y, 0);
                    } else {
                        if ((x >= xio) && (x <= xfo) && (y > yfo)) { //ok - posição 6
                            pos = new Posicao(x, yfo, 0);
                        } else {
                            if ((x > xfo) && (y < yio)) { //ok - posição 7
                                pos = new Posicao(xfo, yio, 0);
                            } else {

```

```

        if ((x > xfo) && ((y >= yio) && (y <= yfo))) { //ok - posição 8
            pos = new Posicao(xfo, y, 0);
        } else {
            pos = new Posicao(xfo, yfo, 0); //ok posicao 9
        }
    }
}
}
}
}
}
}
}
}
}
}
}

```

//verifica se a posição da pessoa mapeada em relação ao objeto é menor que a distância estipulada no objeto

```

    if (!(pos.getX() == x) && (pos.getY() == y)) {
        float distância = (float) Math.sqrt((Math.pow((x - pos.getX()), 2) + Math.pow((y -
pos.getY()), 2)));
        if (distância < (objeto.get(0).getDistância() + p.getRaio())) {
            if (!validadeAviso[i]) {
                String texto = " <JSML> " + "<BREAK MSECs=\"0\"/>" + "<PROS PITCH=\"98\"
RATE=\"200\">"
                + objeto.get(0).getAviso()
                + "</PROS>" + " </JSML> ";
                Sintetizador.speak(texto);
                validadeAviso[i] = true;
            }
        } else {
            validadeAviso[i] = false;
        }
    }
}
}
}
}
}
}
}
}
}

```


}
}
}

Apêndice C

E-mail contendo o valor do Leitor/Receptor RFID

Na figura C.1 vemos o e-mail mandado pela empresa ECHO XIE, importadora dos Leitores/Receptores.

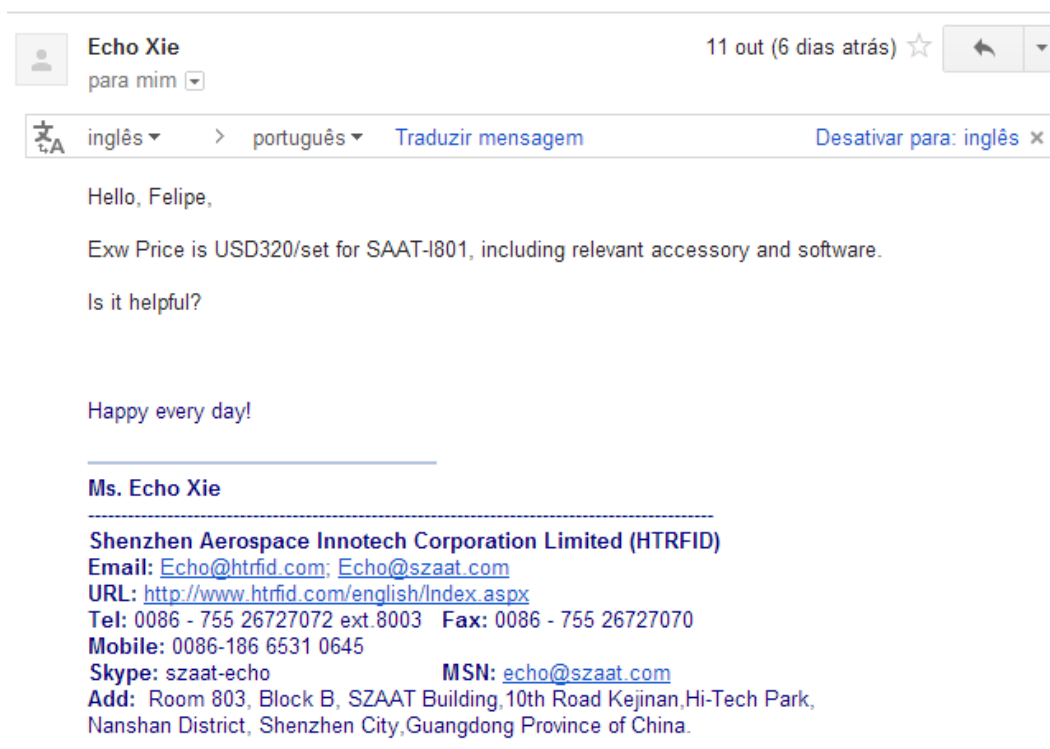


Figura C.1 – E-mail da ECHO XIE referente ao valor do Leitor/Receptor SAAT-I801.

Apêndice D

Telas de Interface do SACCOB

Na figura D.1 é apresentada a tela inicial do sistema SACCOB. Esta interface está dividida em nove partes.

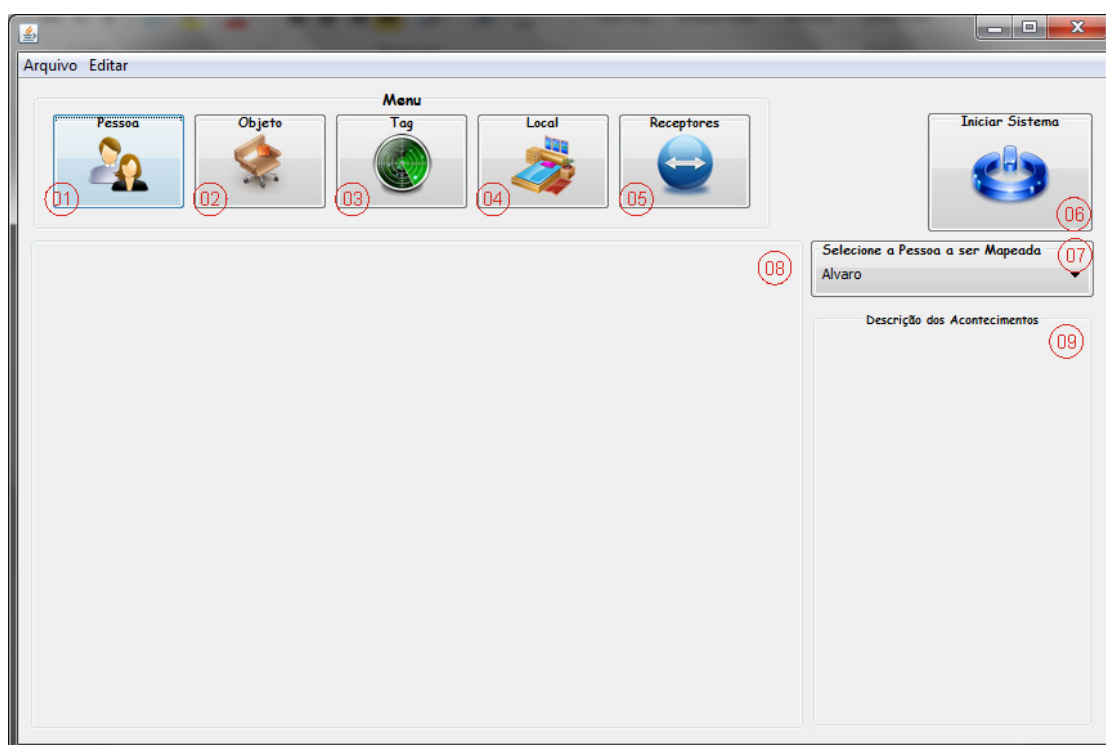


Figura D.1 – Interface inicial do sistema SACCOB.

A seguir será realizada uma explicação sobre cada componente presente na figura D.1.

Parte 01: Menu de Pessoa, que possibilita a visualização de todas as pessoas inseridas no sistema, e apresenta as possibilidades de inserção, edição, busca e remoção de pessoas, como apresentamos na figura D.2.

Parte 02: Menu de Objetos, que possibilita a visualização de todos os objetos inseridos no sistema, e apresenta as possibilidades de inserção, edição, busca e remoção de objetos, como apresentamos na figura D.3.

Parte 03: Menu de *Tags*, que possibilita a visualização de todos as *tags* inseridas no sistema, e apresenta as possibilidades de inserção, edição, busca e remoção de *tags*, como apresentamos na figura D.4.

Parte 04: Menu de Locais, que possibilita a visualização de todos os locais inseridos no sistema, e apresenta as possibilidades de inserção, edição, busca e remoção de locais, como apresentamos na figura D.5.

Parte 05: Menu de Receptores, que possibilita a visualização de todos os receptores inseridos no sistema, e apresenta as possibilidades de inserção, edição, busca e remoção de receptores, como apresentamos na figura D.6.

Parte 06: Botão de inicialização do sistema.

Parte 07: Combo box responsável por seleccionar qual pessoa será mapeada.

Parte 08: Mapa do sistema, onde é possível visualizar os acontecimentos.

Parte 09: Descreve todos os avisos que foram emitidos pelo sistema.

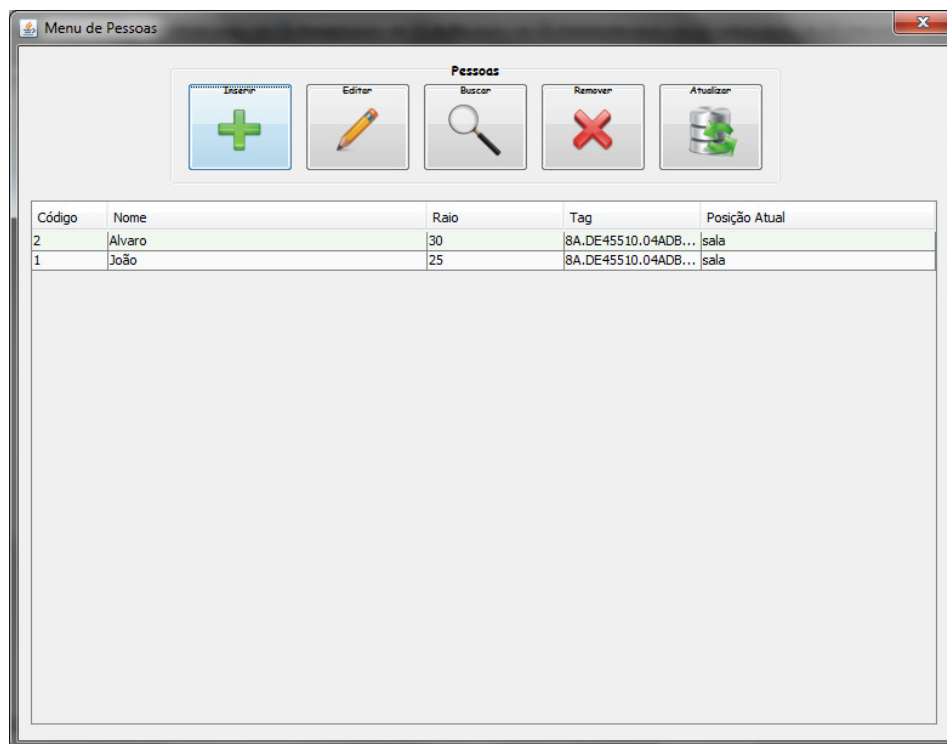


Figura D.2 – Interface Menu de Pessoas do sistema SACCOB.

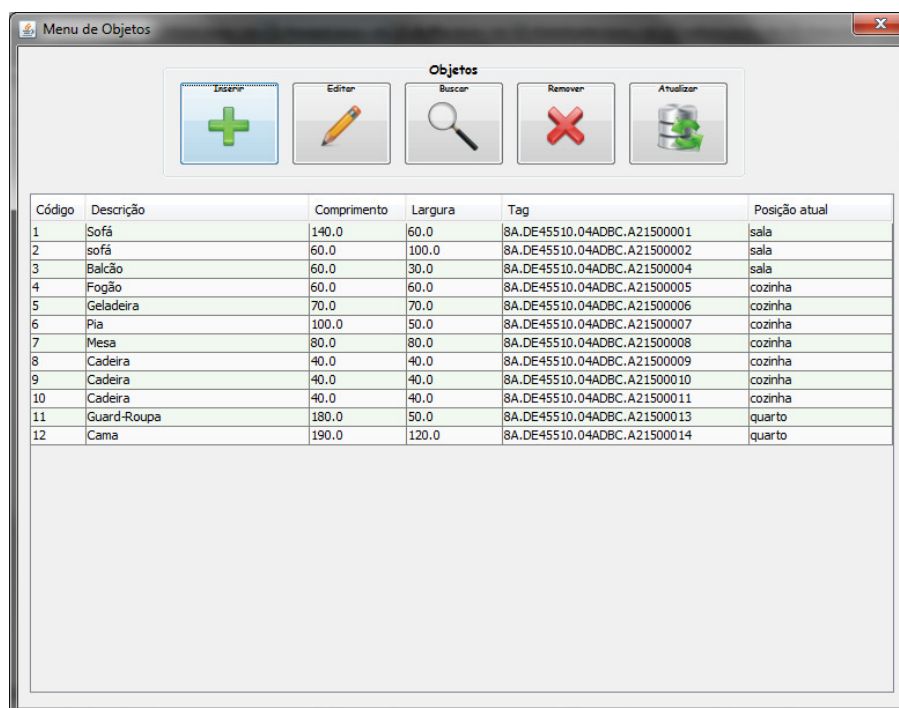


Figura D.3 – Interface Menu de Objetos do sistema SACCOP.

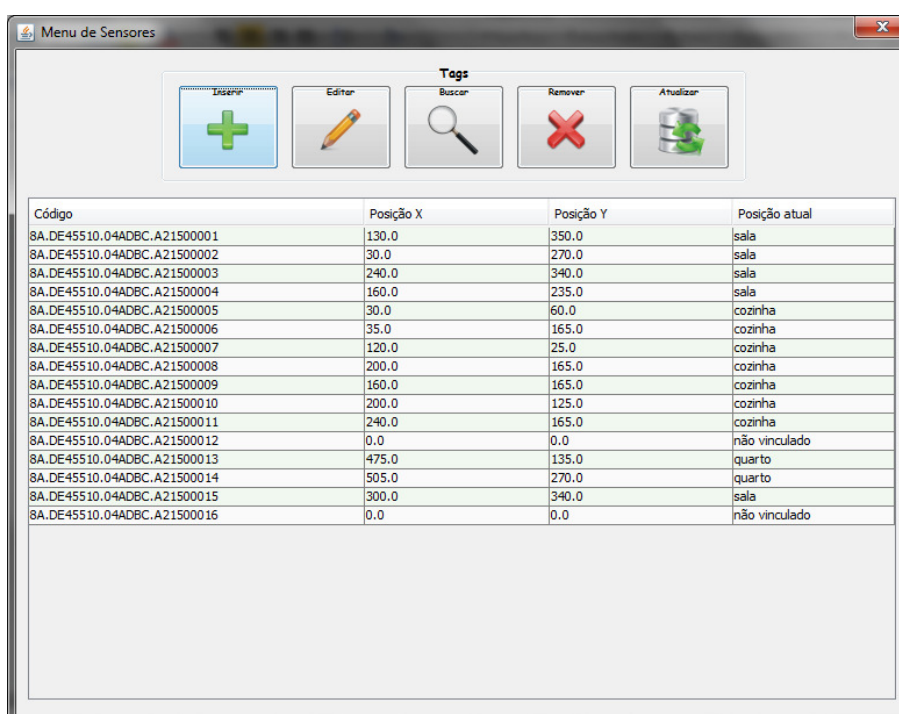


Figura D.4 – Interface Menu de Tags do sistema SACCOP.

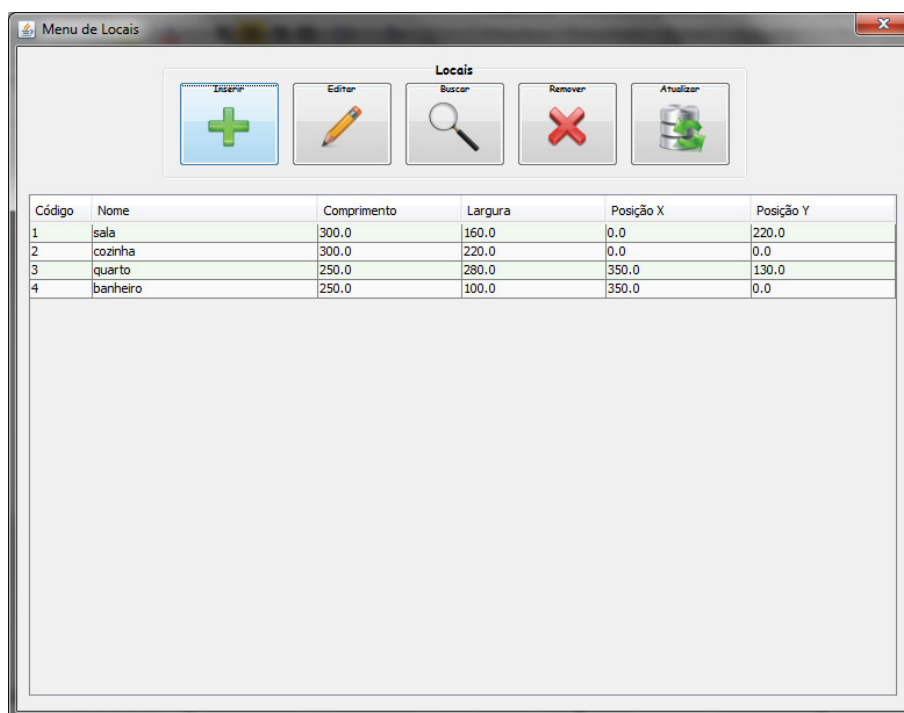


Figura D.5 – Interface Menu de Locais do sistema SACCOP.

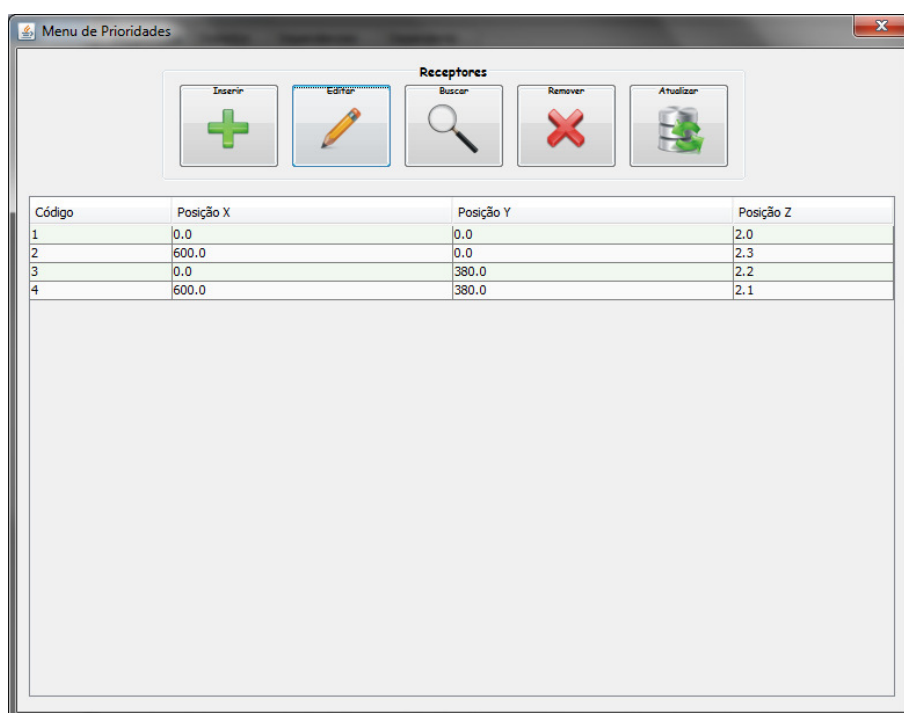


Figura D.6 – Interface Menu de Receptores do sistema SACCOP.

Na figura D.7, o botão “Iniciar Sistema” foi pressionado, e é possível visualizar que o mapa já está visível e mapeado, onde os retângulos vermelhos representam objetos mapeados, os retângulos pretos objetos sem mapeamento (pelo fato de ser um banheiro), os círculos azuis representam pessoas que

estão no ambiente, e o círculo verde representa a pessoa mapeada para a realização de testes de colisão.

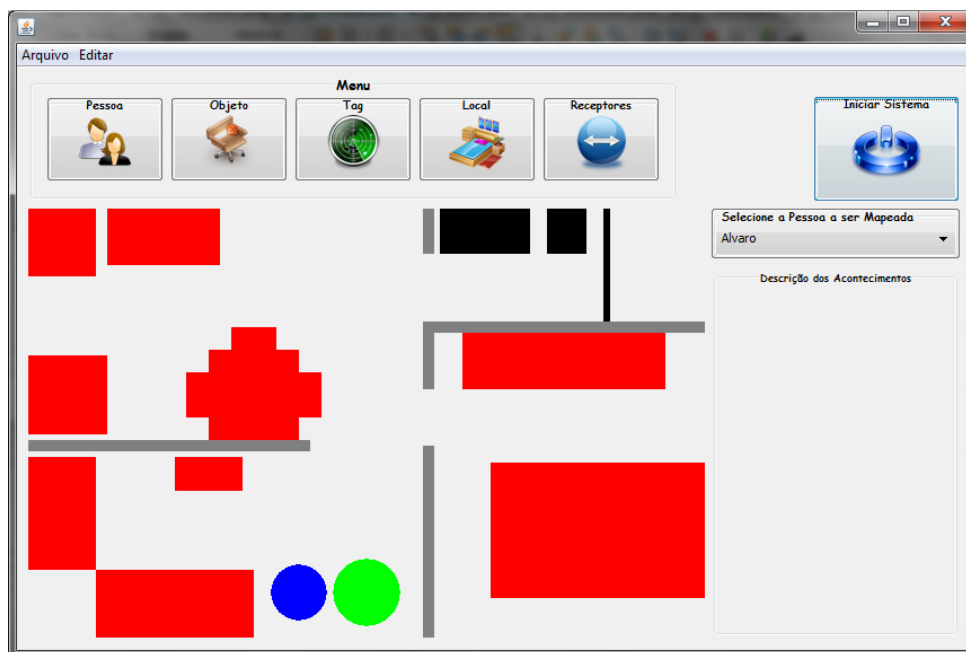


Figura D.7 – Interface com o botão Iniciar Sistema pressionado.

Com o sistema já em funcionamento, cada aviso que é feito para a pessoa mapeada, que é representado pelo círculo verde. Na “Descrição de Acontecimentos” pode ser visualizado os avisos sintetizados para o usuário, e o horário que o ocorreu, apresentados na figura D.8.

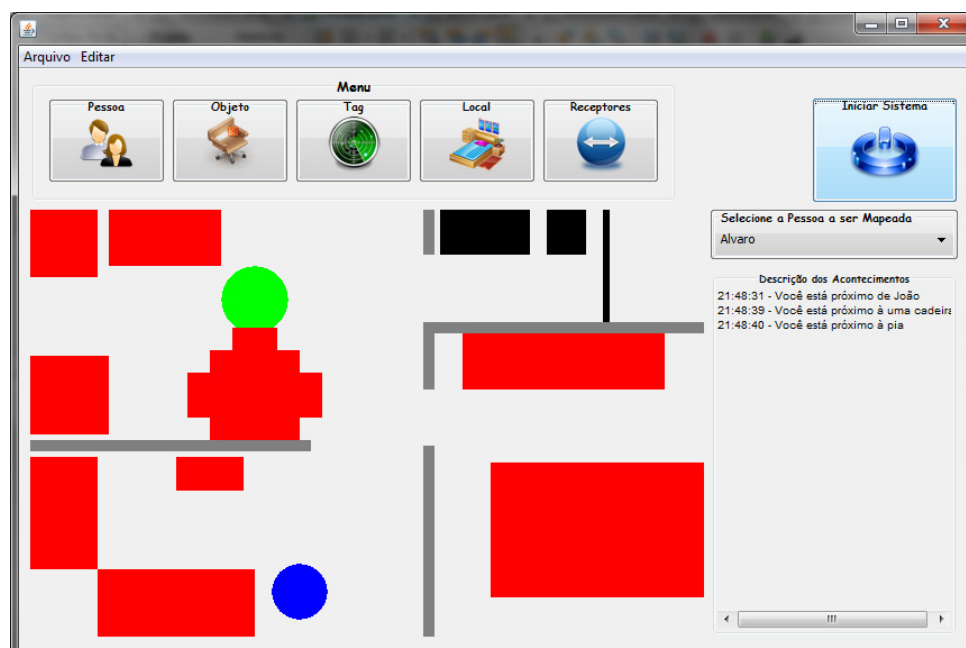


Figura D.8 – Interface com o Sistema em funcionamento.

Referências Bibliográficas

- [1] WHO - World Health Organization. *Visual impairment and blindness*. Consultado na Internet: <http://www.who.int/mediacentre/factsheets/fs282/es/index.html>, em: 23/02/2012.
- [2] IBGE - Instituto Brasileiro de Geografia e Estatística. *Censo 2010*. Consultado na Internet: <http://www.censo2010.ibge.gov.br>, em: 12/07/2012.
- [3] *Censo de 2000*. Consultado na Internet: <http://www.direitoshumanos.gov.br/pessoas-com-deficiencia-1/censo/censo-2000>, em: 23/02/2012.
- [4] BERSCH, R. *Introdução à Tecnologia Assistiva*. Porto Alegre: Centro Especializado em Desenvolvimento Infantil, 2008.
- [5] *DECRETO N° 5.626*. Consultado na Internet: http://www.planalto.gov.br/ccivil_03/_ato2004-2006/2005/decreto/d5626.htm, em: 12/07/2012.
- [6] Instituto Benjamin Constant. *Os Conceitos de Deficiência*. Consultado na Internet: <http://www.ibc.gov.br/?itemid=396>, em: 10/05/2012.
- [7] WHO - World Health Organization. *International classification of impairments, disabilities, and handicaps: a manual of classification relating to the consequences of disease*. Geneva; 1993.
- [8] *Acuidade Visual*. Consultado na Internet: <http://www.vejam.com.br/baixavisao-acuidade-visual>, em: 05/06/2012.
- [9] BRASIL. Ministério da Saúde. Secretaria de Atenção à Saúde. Departamento de Ações Programáticas Estratégicas. *Atenção à saúde da pessoa com deficiência no Sistema Único de Saúde – SUS*. Brasília: Ministério da Saúde, 2009.

- [10] Portaria GM/MS n.º 1060, de 5 de junho de 2002. Consultado na Internet: http://www.saude.mg.gov.br/atos_normativos/legislacao-sanitaria/estabelecimentos-de-saude/reabilitacao/portaria_1060.pdf, em: 25/07/2012
- [11] BRASIL. Ministério da Saúde. *Manual de Legislação em Saúde da Pessoa Portadora de Deficiência*, 2 Ed. Brasília, 2006.
- [12] Governo vai investir R\$ 20 milhões em tecnologias assistivas. Consultado na Internet: http://vidamaislivre.com.br/noticias/noticia.php?id=4497&/governo_vai_investir_r_20_mil_hoes_em_tecnologias_assistivas, em: 11/07/2012.
- [13] JAWS. *JAWS for Windows Screen Reading Software*. Consultado na Internet: <http://www.freedomscientific.com/products/fs/jaws-product-page.asp>, em: 03/07/2012.
- [14] DOSVOX. *Projeto Dosvox*. Consultado na Internet: <http://intervox.nce.ufrj.br/dosvox>, em: 03/07/2012.
- [15] FILHO, J.; VASCONCELOS, F.; MOREIRA, A. *Uso de Robótica Assistiva no Auxílio de Pessoas com Deficiências Visuais*. Consultado na Internet: <http://connepi.ifal.edu.br/ocs/index.php/connepi/CONNepI2010/paper/viewFile/220/193>, em: 05/10/2012.
- [16] ALESSI, A.; PASSOS, I.; RIBEIRO, J. *DOMÓTICA: Bengala Eletrônica via Sensor de Ultrassom*. Brasil: Curitiba, 2010.
- [17] CELESTINO U; ABE J. *Dispositivo Eletrônico Para Auxílio na Locomoção de Deficientes Visuais e/ou Auditivos Baseado na Lógica Paraconsistente Anotada Evidencial*. Consultado na Internet: http://ingepro.com.br/Publ_2011/Agost/455%20pg%2011%20a%2020.pdf, em: 07/10/2012.
- [18] GONZATTO, A.; SANTOS, C.; MELO, F.; RODRIGUES, G.; FARIA, J. *Óculos sonar para deficientes Visuais*. Consultado na Internet: http://www.inicepg.univap.br/cd/INIC_2009/anais/arquivos/RE_0948_0818_01.pdf, em: 07/10/2012.
- [19] JUNIOR, E.; SARNO, T. *Automação de Ambientes para Auxiliar Deficientes Visuais*. Brasil: Goiania, 2008.

- [20] Automatic House. *O que é automação residencial*. Consultado na Internet: <http://www.automatichouse.com.br/AutomaticHouse/WebSite/Automacao/Residencial.aspx>, em: em 12/07/2012.
- [21] AURESIDE - Associação Brasileira de Automação Residencial. *Peculiaridades dos Sistemas de Automação Residencial*. Consultado na Internet: <http://www.aureside.org.br/temastec/default.asp?file=concbasicos02.asp>, em: 12/07/2012.
- [22] DIAS, C.; PIZZOLATO, N. *Domótica: Aplicabilidade e Sistemas de Automação Residencial*. Brasil: Vértices, 2004.
- [23] INTERMEC. *Fundamentos da RFID: Entendendo e usando a identificação por radiofrequência*. Consultado na Internet: <http://www.intermec.com/RFID>, em: 04/07/2012.
- [24] IBM. *Controlando o RFID*. Consultado na Internet: <http://www.ibm.com/br/ibm/ideasfromibm/rfid/061207/index1.phtml>, em 21/11/2012.
- [25] GOMES, G. *Sistema RFID com Tecnologia ZigBee*. Portugal: Lisboa, 2008.
- [26] WARD, M.; KRANENBURG, R.; BACKHOUSE, G. *RFID: Frequency, standards, adoption and innovation*, JISC Technology and Standards Watch, May 2006.
- [27] ISO/IEC 18000-6. *Information technology - Radio frequency identification for item management*. Consultado na Internet: <http://www.youwokeji.com.cn/down/18000-6.pdf>, em: 03/10/2012.
- [28] BOTERO, J.; GONZÁLES, D.; PAZ, H. La interferência como un fator que afecta el desempeño de um sistema RFID. *Red de Revistas Científicas de América Latina, el Caribe, España y Portugal*. Colômbia, 2009.
- [29] AEROSPACE INNOTECH. *RFID Middle Range Integrated Reader SAAT-I801*. Consultado na Internet: http://www.htrfid.com/english/product_wireless_show.aspx?pk=125, em: 15/09/2012.
- [30] ALIBABA. *Long Range Passive RFID Tag*. Consultado na Internet: http://www.alibaba.com/product-gs/635166429/long_range_passive_rfid_tag.html, em: 28/09/2012.
- [31] BECK K. *Extreme Programming Explained: Embrace Change*. Reading, Massachusetts: Addison Wesley Longman, Inc., 1999.

- [32] SCHWABER, K.; BEEDLE, M. *Agile Software Development with SCRUM*. Prentice Hall, 2002.
- [33] BUSCHMANN, F.; MEUNIER, R.; ROHNERT, H.; SOMMERLAD, P.; STAL, M. *Pattern-Oriented Software Architecture: A System of Patterns*. New York: John Wiley & Sons, 1996.
- [34] ORACLE. *Java Speech API*. Consultado na Internet: <http://java.sun.com/products/java-media/speech>, em: 30/02/2012.
- [35] POSTGRESQL. *PostgreSQL Documentation*. Consultado na Internet: <http://www.postgresql.org/docs/>, em: 30/02/2012.
- [36] ALUR, D.; MALKS, D.; CRUPI, J. *Core J2EE Patterns: Best Practices and Design Strategies*, 2 Ed. California: Sun Microsystems, 2003.
- [37] GAMMA, E.; HELM, R.; JOHNSON, R.; VLISSIDES, J. *Design Patterns: Elements of Reusable Object-Oriented Software*, Addison-Wesley, 1995.
- [38] SILBERSCHATZ, A. *Sistemas Operacionais - Conceitos e Aplicações*. 7. ed. Reading: Editora Campus, 2008.
- [39] HALLIDAY, D; RESNICK, R; WALKER, J. *Fundamentos da Física: Gravitação, Ondas e Termodinâmica*. LTC S.A., Rio de Janeiro, 2003.
- [40] MALTA, S; SALES, J; MARTINS, M; RODRIGUES, A. *Medição de blindagem eletromagnética adaptada à realidade dos laboratórios de Eletromagnetismo*. Belo Horizonte: CEFET-MG, 2009.
- [41] STEINBRUCH, A.; WINTERLE, P. *Geometria Analítica*, 2. ed. São Paulo: Pearson Makron Books, 1987.