

TRABALHO DE GRADUAÇÃO

**MONITORAMENTO DE AMBIENTES COM DISTINÇÃO
DE USUÁRIOS USANDO RFID PASSIVO APLICADO
A AMBIENTES INTELIGENTES**

Leonardo da Fonseca Pereira

Kenneth Lui

Brasília, Dezembro de 2020



**ENGENHARIA
MECATRÔNICA**
UNIVERSIDADE DE BRASÍLIA

UNIVERSIDADE DE BRASÍLIA
Faculdade de Tecnologia
Curso de Graduação em Engenharia de Controle e Automação

TRABALHO DE GRADUAÇÃO

MONITORAMENTO DE AMBIENTES COM DISTINÇÃO
DE USUÁRIOS USANDO RFID PASSIVO APLICADO
A AMBIENTES INTELIGENTES

Leonardo da Fonseca Pereira

Kenneth Lui

*Relatório submetido como requisito parcial de obtenção
de grau de Engenheiro de Controle e Automação*

Banca Examinadora

Prof. Adolfo Bauchspiess, ENE/UnB

Orientador

Lélio Ribeiro Soares Júnior

Examinador interno

Daniel Chaves Café

Examinador interno

Brasília, Dezembro de 2020

FICHA CATALOGRÁFICA

PEREIRA, LEONARDO F., LUI, KENNETH.

Monitoramento de ambientes com distinção de usuários usando RFID passivo aplicado a ambientes inteligentes,

[Distrito Federal] 2020.

x, 149p., 297 mm (FT/UnB, Engenheiro, Controle e Automação, 2020). Trabalho de Graduação – Universidade de Brasília. Faculdade de Tecnologia.

1. RFID

2. Automação predial

3. IoT

4. Eficiência energética

I. Mecatrônica/FT/UnB

II. Título (Série)

REFERÊNCIA BIBLIOGRÁFICA

PEREIRA, LEONARDO F., LUI, KENNETH, (2020). Monitoramento de ambientes com distinção de usuários usando RFID passivo aplicado a ambientes inteligentes. Trabalho de Graduação em Engenharia de Controle e Automação, Publicação FT.TG-*n*°03, Faculdade de Tecnologia, Universidade de Brasília, Brasília, DF, 149p.

CESSÃO DE DIREITOS

AUTORES: Leonardo da Fonseca Pereira e Kenneth Lui

TÍTULO DO TRABALHO DE GRADUAÇÃO: Monitoramento de ambientes com distinção de usuários usando RFID passivo aplicado a ambientes inteligentes.

GRAU: Engenheiro

ANO: 2020

É concedida à Universidade de Brasília permissão para reproduzir cópias deste Trabalho de Graduação e para emprestar ou vender tais cópias somente para propósitos acadêmicos e científicos. O autor reserva outros direitos de publicação e nenhuma parte desse Trabalho de Graduação pode ser reproduzida sem autorização por escrito do autor.

Leonardo da Fonseca Pereira

Kenneth Lui

SQN 103, Bloco B, Apto 403.

QNB 11, Casa 22

70732-020 Brasília – DF – Brasil.

72115-110 Brasília – DF – Brasil.

Dedicatórias

Dedico este trabalho a todas as pessoas que nos ajudaram durante a execução do projeto.

Kenneth Lui

Dedico este trabalho à minha família, amigos e namorada, que me apoiaram durante este projeto.

Leonardo da Fonseca Pereira

Agradecimentos

Agradeço primeiramente à toda minha família, que sempre estiveram comigo e me deram todo o apoio e suporte durante todos esses anos de graduação, principalmente nos momentos mais difíceis.

Agradeço aos meus amigos de curso e namorada, que me auxiliaram e motivaram durante todo este percurso.

Agradeço à empresa júnior Mecajun, que me propiciou crescimento profissional e pessoal, além de grandes amizades que levarei para a vida inteira.

Agradeço ao Thulio, que ajudou na execução dos vídeos gravados.

Agradeço à todos que ajudaram na realização dos experimentos e testes deste projeto.

Agradeço ao professor Adolfo por sua orientação, paciência e pela oportunidade de realizar este projeto.

Leonardo da Fonseca Pereira

Agradeço o apoio de toda a minha família, sem eles eu não estaria hoje na Universidade de Brasília.

Agradeço o André Almeida, cujo trabalho de graduação serviu de inspiração para o nosso projeto. Além disso nos auxiliou nas fases iniciais do trabalho.

Agradeço aos meus amigos que nos ajudaram durante a realização de testes para a validação do projeto.

Agradeço ao professor e orientador Adolfo por todo o suporte e apoio durante o trabalho de graduação.

Kenneth Lui

RESUMO

Este trabalho propõe o desenvolvimento de um sistema para realizar o monitoramento de ambientes com a individualização das tags associadas aos usuários usando RFID passivo, de modo a conseguir personalizar o cálculo de carga térmica com base nos diferentes usuários dentro dos ambientes. Através desses dados, foi utilizado um arduino para realizar o acionamento de relés dependendo da quantidade de pessoas em cada ambiente.

O sistema foi desenvolvido com base em 3 ambientes do Laboratório de Automação e Robótica, LARA/UnB, assim como o ambiente externo. Foram utilizadas 2 antenas e 1 leitora da empresa Impinj para cada ambiente, totalizando 3 leitoras e 6 antenas. A identificação do usuário e a transição do mesmo entre os ambientes é feita através da sua passagem por duas antenas, que ficam entre a divisória dos ambientes. Dessa forma, analisando o valor de RSSI e o tempo da transição obtido da leitura das antenas é possível identificar a transição de um usuário para outro ambiente.

O sistema foi desenvolvido utilizando kit de desenvolvimento Octane SDK disponibilizado pela Impinj. Com ele foram feitos programas utilizando uma linguagem orientada a objetos C#. Foram analisados parâmetros como distância, velocidade e orientação da TAG com relação às leitoras.

Palavras Chave: RFID, Monitoramento de ambientes, Rastreamento de pessoas, Automação predial, IoT, Eficiência energética, conforto térmico

ABSTRACT

This work proposes the development of a system to perform the monitoring of environments with the individualization of the tags associated to the users using passive RFID, in order to be able to customize the calculation of thermal load based on the different users within the environments. Through these data, an arduino was used to trigger relays depending on the number of people in each environment.

The system was developed based on 3 environments of the Automation and Robotics Laboratory, LARA/UnB, as well as the external environment. 2 antennas and 1 reader from the company Impinj were used for each environment, totaling 3 readers and 6 antennas. The identification of the user and his transition between the environments is done through his passage through two antennas, which are between the partition of the environments. Thus, by analyzing the RSSI value and the transition time obtained from reading the antennas, it is possible to identify the transition from a user to another environment.

The system was developed using the Octane SDK development kit provided by Impinj. With it, programs were made using a language oriented to objects C#. Parameters such as distance, speed and orientation of the TAG in relation to the readers were analyzed.

Keywords: RFID, Environment monitoring, People tracking, Building automation, IoT, Energy efficiency, thermal comfort

SUMÁRIO

1	Introdução	1
1.1	CONTEXTUALIZAÇÃO	1
1.2	DEFINIÇÃO DO PROBLEMA	2
1.3	OBJETIVOS DO PROJETO	2
1.4	TRABALHOS ANTERIORES	3
1.5	APRESENTAÇÃO DO MANUSCRITO	3
2	Fundamentação Teórica	4
2.1	CONFORTO TÉRMICO	4
2.2	TERMORREGULAÇÃO HUMANA	5
2.3	PVM: ÍNDICE DE CONFORTO TÉRMICO	6
2.4	CARGA TÉRMICA	7
2.5	RFID	8
2.6	ANTENAS	9
2.7	TAGS	9
2.8	EPC	10
2.9	LEITORAS	10
2.10	MIDDLEWARE	10
2.11	RSSI	11
2.12	EFEITO DOPPLER	12
2.13	MODOS DE LEITURA DA IMPINJ SPEEDWAY REVOLUTION R420	12
2.14	RELAÇÃO ENTRE ÁREA DE LEITURA E SENSIBILIDADE	14
2.15	TRADE-OFF ENTRE TAXA DE LEITURA E RESISTÊNCIA À INTERFERÊNCIAS	15
3	Materiais e Ferramentas	17
3.1	VISUAL STUDIO	17
3.2	WINDOWS FORMS	17
3.3	LEITORAS IMPINJ SPEEDWAY REVOLUTION R420	18
3.4	ANTENA IMPINJ THRESHOLD	19
3.5	TAG RFID UHF ADESIVA 915 - 868 MHZ HIGGS3 PASSIVA	20
3.6	ITEMTEST	20
3.7	OCTANE SDK	21
3.8	ROTEADOR TP-LINK	22

3.9	ARDUINO	23
3.10	RELÉ	24
3.11	LÂMPADA LED	25
4	Experimentos e Resultados	26
4.1	INTRODUÇÃO	26
4.2	POSICIONAMENTO DAS LEITORAS	26
4.3	DETECÇÃO DE PESSOAS USANDO INFORMAÇÕES DO EFEITO DOPPLER.....	30
4.3.1	EXPERIMENTO UTILIZANDO VALOR MÍNIMO DE 4 HZ DE FREQUÊNCIA DOPPLER PARA CONSIDERAR PASSAGEM	31
4.3.2	REPETIÇÃO DO EXPERIMENTO	34
4.4	DETECÇÃO DE PESSOAS USANDO INFORMAÇÕES DO RSSI	35
4.4.1	EXPERIMENTO DE DETECÇÃO DA TAG UTILIZANDO VALOR MÍNIMO DE -47 dBm DE RSSI PARA CONSIDERAR PASSAGEM.....	36
4.5	TESTES DE TRANSIÇÕES ENTRE AMBIENTES.....	40
4.6	ACIONAMENTO DO AR CONDICIONADO	43
4.7	INTERFACE GRÁFICA	47
4.8	TESTE DE MOVIMENTAÇÃO DE TAG COM PARADA NO CAMINHO.....	51
4.9	PASSAGEM DE 2 TAGS SIMULTANEAMENTE.....	53
4.10	PASSAGEM DE 3 TAGS SIMULTANEAMENTE.....	55
4.11	PASSAGEM DE 4 TAGS SIMULTANEAMENTE.....	58
4.12	PASSAGEM DE 5 TAGS SIMULTANEAMENTE.....	60
4.13	DUAS PASSAGENS SIMULTÂNEAS EM DOIS AMBIENTES DIFERENTES	63
4.14	TESTES DE LEITURA DE ACORDO COM DISTÂNCIA	66
4.14.1	TESTES COM TAG ORIENTADA COM A FACE PARA A ANTENA.....	66
4.14.2	TESTES COM TAG ORIENTADA PERPENDICULARMENTE À ANTENA	70
5	Análise dos Resultados.....	76
5.1	RESUMO DOS RESULTADOS	76
5.2	VISÃO GERAL DOS RESULTADOS OBTIDOS NO PROJETO.....	81
6	Conclusões.....	82
6.1	PERSPECTIVAS FUTURAS.....	83
	REFERÊNCIAS BIBLIOGRÁFICAS	85
	Anexos.....	88
I	Programas utilizados.....	89
II	Manual de Instalação e Configuração.....	122
II.1	OPENHAB	122
II.1.1	PRE-REQUISITOS.....	122
II.1.2	INSTALAÇÃO OPENHAB	122

II.1.3	INSTALAÇÃO MOSQUITTO MQTT BROKER	123
II.2	ARDUINO	126
II.2.1	ESP8266	126
II.3	INSTALAÇÃO E TESTE DAS LEITORAS E ANTENAS.....	134
II.3.1	INSTALAÇÃO FÍSICA	134
II.3.2	CONFIGURAÇÃO DA REDE	137
II.3.3	TESTE DO SISTEMA	141

LISTA DE FIGURAS

2.1	Representação dos fatores externos que influenciam o conforto térmico. Fonte: https://ggbec.co.uk/quick-guide-overheating-thermal-comfort	5
2.2	Representação da fisiologia humana levando em conta as trocas térmicas. [1].....	6
2.3	PPD em função do PPD [1].....	7
2.4	Exemplo de sistema RFID. Fonte: https://www.hellermannntyton.com.br/	8
2.5	Componentes básicos de uma tag. Fonte: https://rfidmoura.wordpress.com/	9
2.6	Exemplo de sistema de RFID com middleware [2].	11
2.7	Fonte: Página do site da Impinj sobre modos de leitura	12
2.8	Fonte: Página do site da Impinj sobre modos de leitura	14
2.9	Fonte: Página do site da Impinj sobre modos de leitura	15
3.1	Ilustração da leitora Impinj Speedway Revolution R420.	18
3.2	Exemplo de rede configuração de rede utilizando cabo ethernet e switch de rede. Imagem retirada de [3].	19
3.3	Imagem da antena Impinj Threshold. Imagem retirada de [4].....	19
3.4	Portas da antena no Reader Speedway R420. Imagem retirada de [3].....	20
3.5	Tags RFID passivas utilizadas no projeto.	20
3.6	Interface do software ItemTest. Imagem retirada de [5].	21
3.7	Ilustração da parte frontal do roteador.	22
3.8	Ilustração da parte traseira do roteador.	23
3.9	Ilustração do arduino Uno.	24
3.10	Ilustração do relé.	25
3.11	Ilustração da lâmpada LED utilizada.	25
4.1	Disposição das leitoras e antenas no LARA.	27
4.2	Legenda da figura 4.1.....	27
4.3	Planta da casa onde foi implementado o projeto.	28
4.4	Imagem geral da casa.....	29
4.5	Imagem da sala de reuniões (esquerda) e do corredor de baias (direita).	29
4.6	Imagem da área externa.	30
4.7	Exemplo de leitura de passagem de uma tag por uma antena.	31
4.8	Leitura da frequência Doppler para a antena 1.	31
4.9	Contagem da detecção de passagens pela antena 1.	32
4.10	Leitura da frequência Doppler para a antena 2.	33

4.11	Contagem da detecção de passagens pela antena 2.	33
4.12	Exemplo de leitura do valor RSSI na passagem de uma tag por uma antena.	35
4.13	Leitura do valor RSSI na passagem da tag pela antena 1 dez vezes.	36
4.14	Contagem da detecção de passagens pela antena 1.	37
4.15	Leitura do valor RSSI na passagem da tag pela antena 2 dez vezes.	37
4.16	Contagem da detecção de passagens pela antena 2.	38
4.17	Leituras das 10 passagens pela antena 1 e antena 2.	39
4.18	Exemplo de 3 transições da Sala Principal para o Corredor de Baias.	40
4.19	Experimento 1 para detecção de passagem de tags entre ambientes.	41
4.20	Experimento 2 para detecção de passagem de tags entre ambientes.	42
4.21	Experimento 3 para detecção de passagem de tags entre ambientes.	42
4.22	Experimento 4 para detecção de passagem de tags entre ambientes.	42
4.23	Experimento 5 para detecção de passagem de tags entre ambientes.	43
4.24	Experimento 6 para detecção de passagem de tags entre ambientes.	43
4.25	Experimento 1 para acionamento do LED em cada ambiente.	44
4.26	Experimento 2 para acionamento do LED em cada ambiente.	45
4.27	Experimento 3 para acionamento do LED em cada ambiente.	45
4.28	Experimento 4 para acionamento do LED em cada ambiente.	45
4.29	Experimento 5 para acionamento do LED em cada ambiente.	45
4.30	Experimento 6 para acionamento do LED em cada ambiente.	46
4.31	Tela inicial da interface sem pessoas dentro dos ambientes.	47
4.32	Interface após transição de uma pessoa do ambiente externo para a sala principal. ..	48
4.33	Interface com duas pessoas na sala principal.	48
4.34	Interface com três pessoas na sala principal.	49
4.35	Interface com a transição de uma pessoa da sala principal para o corredor de baias. ..	49
4.36	Interface com a transição de duas pessoas da sala principal para o corredor de baias. ..	50
4.37	Interface com a transição de três pessoas da sala principal para o corredor de baias. ..	50
4.38	Experimento de movimentação de tag com parada no caminho.	51
4.39	Gráfico do número de pessoas dentro do ambiente externo durante o experimento. ..	52
4.40	Gráfico do número de pessoas dentro da sala principal durante o experimento.	52
4.41	Leitura RSSI para passagem de 2 tags simultaneamente.	53
4.42	Contagem de número de pessoas na área externa.	54
4.43	Contagem de número de pessoas na sala principal.	54
4.44	Acionamento do rele na sala principal.	55
4.45	Leitura RSSI para passagem de 3 tags simultaneamente.	56
4.46	Contagem de número de pessoas na área externa.	56
4.47	Contagem de número de pessoas na sala principal.	57
4.48	Acionamento do relé na sala principal.	57
4.49	Leitura RSSI para passagem de 4 tags simultaneamente.	58
4.50	Contagem de número de pessoas na área externa.	59
4.51	Contagem de número de pessoas na sala principal.	59
4.52	Acionamento do rele na sala principal.	60

4.53	Leitura RSSI para passagem de 5 tags simultaneamente.	61
4.54	Contagem de número de pessoas na área externa.	61
4.55	Contagem de número de pessoas na sala principal.	62
4.56	Acionamento do rele na sala principal.	62
4.57	Número de pessoas na sala principal.	64
4.58	Número de pessoas na sala de reuniões.	64
4.59	Nº de pessoas no corredor de baias.	65
4.60	Ilustração da orientação da tag na passagem com a face voltada para a antena.	66
4.61	Passagem a 30 cm de distância da antena com a tag orientada para a antena.	67
4.62	Passagem a 30 cm de distância da antena com a tag orientada para a antena.	67
4.63	Passagem a 60 cm de distância da antena com a tag orientada para a antena.	68
4.64	Passagem a 60 cm de distância da antena com a tag orientada para a antena.	68
4.65	Passagem a 90 cm de distância da antena com a tag orientada para a antena.	69
4.66	Passagem a 90 cm de distância da antena com a tag orientada para a antena.	69
4.67	Ilustração da orientação da tag na passagem.	70
4.68	Passagem a 30 cm de distância da antena com a tag orientada perpendicularmente à antena.	71
4.69	Passagem a 30 cm de distância da antena com a tag orientada perpendicularmente à antena.	71
4.70	Passagem a 60 cm de distância da antena com a tag orientada perpendicularmente à antena.	72
4.71	Passagem a 60 cm de distância da antena com a tag orientada perpendicularmente à antena.	72
4.72	Passagem a 90 cm de distância da antena com a tag orientada perpendicularmente à antena.	73
4.73	Passagem a 90 cm de distância da antena com a tag orientada perpendicularmente à antena.	73
4.74	Passagem a 45 cm de distância da antena com a tag orientada perpendicularmente à antena.	74
4.75	Passagem a 45 cm de distância da antena com a tag orientada perpendicularmente à antena.	75
5.1	Tabela resumo com a comparação entre as leituras usando Efeito Doppler e RSSI. ..	76
5.2	Tabela resumo com o resultado dos experimentos com transições sequenciais.	77
5.3	Resultado do experimento dos acionamentos dos LEDs.	78
5.4	Tabela resumo do resultado dos experimentos com passagens simultâneas.	79
5.5	Tabela resumo do resultado dos experimentos variando a distância entre a tag e a antena, com a tag orientada com a face para a antena.	79
5.6	Tabela resumo do resultado dos experimentos variando a distância entre a tag e a antena, com a tag orientada com a face perpendicularmente à antena.	80
II.1	Pinagem do módulo ESP-01	127
II.2	Circuito e conexão dos pinos do módulo ESP-01 para modo programação	128

II.3	Configuração do programa Termite	129
II.4	Comando e resposta do programa Termite.....	129
II.5	Janelas que abrem ao executar ESP Flash Download Tool	130
II.6	Endereços ADDR ao lado de seus respectivos arquivos.....	132
II.7	Resultado na janela log ao fim do upgrade	133
II.8	Portas E/S localizadas no Reader Speedway R420. Imagem retirada de [3].	134
II.9	Portas da antena e LEDs no Reader Speedway R420. Imagem retirada de [3].	135
II.10	Exemplo de rede configuração de rede utilizando cabo ethernet e switch de rede. Imagem retirada de [3].	137
II.11	Configurações da rede Wi-Fi.....	138
II.12	Configuração do IP fixo.	138
II.13	Teste da rede utilizando a função "ping"no prompt do computador.	139
II.14	Usuário e senha para acessar IP do Reader.	139
II.15	Interface web do Reader.	140
II.16	Tela inicial do aplicativo ItemTest da Impinj.	141
II.17	Tela de configurações do Reader.	142
II.18	Configuração de um novo Reader.....	142
II.19	Configuração da forma de leitura do Reader.	143
II.20	Reader habilitado na aba de "Reader Settings".	144
II.21	Leitura das TAGs no ItemTest.....	145
II.22	TAGs fora do campo de leitura do Reader.	146
II.23	Procura pelo Octane SDK no Visual Studio. Imagem retirada de [6].....	146
II.24	Conteúdo do pacote no Visual Studio. Imagem retirada de [6].	147
II.25	Botão de instalar do Octane SDK no Visual Studio. Imagem retirada de [6].	147
II.26	Aceite os termos da licença para iniciar a instalação no Visual Studio. Imagem retirada de [6].....	148
II.27	Mensagem indicando que a instalação foi um sucesso. Imagem retirada de [6].	148
II.28	Altere o IP do Reader no código do Reader Connectivity.....	149
II.29	Leitura de TAGs através do código exemplo "ReaderConnectivity"no Visual Studio.	149

LISTA DE TABELAS

2.1	Padrão ASHRAE	7
4.1	Tabela resumo do resultado do experimento.....	34
II.1	Padrões do LED de power. Informações retiradas de [3].	135
II.2	Padrões do LED do status do Reader. Informações retiradas de [3].	136
II.3	Padrões do LED do status do antena. Informações retiradas de [3].	136

LISTA DE SÍMBOLOS

Siglas

PCI	<i>Peripheral Component Interconnect</i>
CPU	Unidade Central de Processamento - <i>Central Processing Unit</i>
AO	Saída Analógica - <i>Analog Out</i>
DO	Saída Digital - <i>Digital Out</i>
CS	Seletor de <i>Chip</i> - <i>Chip Select</i>
SC	Sem Conexão
P.I.	Placa de Interface
ICW	<i>Initialization Command Words</i>
OCW	<i>Operational Control Word</i>
EPE	<i>Empresa de Pesquisa Energética</i>
EPC	<i>Electronic Product Code</i>
SDK	<i>Software Development Kit</i>
RFID	<i>Radio-Frequency IDentification</i>
RSSI	<i>Received Signal Strength Indicator</i>
PPD	<i>Predicted Percentage of Dissatisfied</i>
PMV	<i>Predicted Mean Vote</i>
LED	<i>Light Emitting Diode</i>
IP	<i>Internet Protocol</i>
ASHRAE	<i>American Society of Heating, Refrigerating and Air-Conditioning Engineers</i>
USB	<i>Universal Serial Bus</i>
IDE	<i>Integrated Development Environment</i>
POE	<i>Power Over Ethernet</i>
LARA	<i>Laboratório de Automação e Robótica</i>
SG11	<i>Laboratório de Engenharia Elétrica da UnB</i>
COVID-	<i>Coronavírus</i>

Capítulo 1

Introdução

1.1 Contextualização

No ano de 2017 foi feita uma consulta pública no Brasil para medir os níveis mínimos de eficiência energética do ar condicionado. O resultado mostrou que, apesar da eficiência energética ter melhorado nos últimos anos, ainda estamos muito abaixo da média de países que se preocupam com este assunto, como por exemplo o Japão e Coreia que são países líderes na área.

Um aparelho de ar condicionado comum consome em média 25 KWh/mês para ficar ligado uma hora por dia durante um mês. Se ele ficar ligado quatro horas por dia durante o mês, consumirá 100KWh/mês. De acordo com o Anuário Estatístico de Energia Elétrica 2018 da EPE, a residência brasileira consome em média 157,9 KWh/mês. Isso significa que o ar condicionado, se ficasse ligado 4 horas por dia durante um mês, gastaria o equivalente a aproximadamente 63% do consumo energético dessa residência.

A mesma consulta pública de 2017 mostrou que o consumidor comum seria capaz de comprar um ar condicionado novo com o dinheiro economizado caso a eficiência energética do Brasil fosse similar a de países como Japão e Coreia. A nível nacional, o aumento na eficiência energética do ar condicionado se traduziria na diminuição de centenas de milhares de reais gastos com energia elétrica e até poderia evitar a construção de novas usinas hidrelétricas.

Somado a isso, cada vez mais se fala em automação predial. A integração de sistemas de ar-condicionados, segurança, iluminação e outros sistemas possibilita uma maior praticidade, conforto e eficiência. Diferentes tecnologias são aplicadas atualmente para se atender esse propósito, e uma delas é a identificação por radiofrequência.

1.2 Definição do problema

Embora atualmente as pessoas estejam mais cientes do alto consumo energético do ar condicionado, ainda não temos muitos métodos para contornar a situação. Ainda estamos longe de ter equipamentos com a mesma eficiência energética de países como Coreia e Japão e por enquanto a principal forma de usá-los de forma sustentável é depender do bom senso do consumidor.

1.3 Objetivos do projeto

O objetivo deste trabalho é utilizar a tecnologia de identificação por radiofrequência para realizar o monitoramento de ambientes e rastreamento por proximidade de pessoas de forma a criar ambientes inteligentes e usar o ar condicionado de forma alternativa, propiciando uma melhor eficiência energética e mantendo o conforto térmico do usuário de forma sustentável.

Seria inviável, além de complexo, reestruturar o projeto do ar condicionado para melhorar a eficiência energética. Portanto, a solução encontrada foi monitorar e controlar o ambiente onde o equipamento está sendo utilizado.

Por meio de equipamentos que utilizam RFID passivo para identificação de tags é possível identificar o número de pessoas que estão dentro de cada ambiente, obtendo informações individuais de cada pessoa que está associada a tag, como gênero, peso e altura.

As informações obtidas tornam possível o controle do ar condicionado de forma antecipativa, ou seja, o equipamento se anteciparia à variação de temperatura interna de forma a ligar e desligar automaticamente, e apenas pelo tempo e intensidade necessário, com base nas informações coletadas. Isso significa que o ar condicionado não precisaria estar sempre ligado na potência máxima e estaria ativo em apenas uma fração do tempo que estaria caso não houvesse nenhum tipo de controle sobre ele.

1.4 Trabalhos anteriores

Outros trabalhos envolvendo RFID já foram desenvolvidos anteriormente no LARA/UnB, como é o caso do trabalho de Oliveira, Filipe e Rocha, Frederico [7], do trabalho de Alves, Raissa e Chupel, Renata [8] e do trabalho de Almeida, André [9]. O trabalho de Rodrigues, Alexandre e Silva, Mariana [10] desenvolveu um controle antecipativo utilizando visão computacional. O presente trabalho dá continuidade a esses projetos buscando contribuir através de uma nova abordagem de rastreamento de pessoas por proximidade.

1.5 Apresentação do manuscrito

Nas próximas seções será apresentado toda a fundamentação teórica por trás do projeto para que seja possível compreender o que foi feito.

No capítulo 2 foi explicado os fenômenos físicos utilizados para a compreensão e análise da passagem de tags, que representam a entrada e saída de pessoas em um determinado ambiente.

No capítulo 3 foi enumerado os programas e ferramentas utilizados durante todo o processo e explicado a forma como devem ser configuradas para que este projeto possa ser facilmente replicado.

No capítulo 4 foi documentado os resultados dos testes e cenários criados de forma a avaliar o funcionamento deste projeto.

Por fim, no capítulo 5 foi feita a conclusão do projeto, explanando os resultados obtidos e mostrando uma visão geral do trabalho, assim como perspectivas futuras.

Capítulo 2

Fundamentação Teórica

2.1 Conforto Térmico

O conforto térmico é uma variável subjetiva de difícil conceituação, visto que não se limita a fatores físicos ou calculáveis. Este fator subjetivo do conforto térmico se dá por causa da dependência de fatores pessoais que variam de indivíduo para indivíduo.

Sensações térmicas diferentes são percebidas por diferentes pessoas mesmo estando em um ambiente em comum. Diferentemente de sensores de temperatura que para uma determinada entrada terá sempre a mesma saída, o corpo humano não responde da mesma forma. Na verdade, pode-se dizer que pessoas submetidas à mesma temperatura e clima podem ter sensações completamente diferentes em relação ao conforto térmico.

Para Hensen, conforto térmico pode ser definido como um estado no qual não há impulso de condução para corrigir o comportamento do ambiente [11]. Por outro lado, a ASHARE define o conforto térmico como a condição da mente em que existe satisfação com o ambiente térmico [12]. Deste modo, a satisfação da pessoa será influenciada dependendo de diversos fatores, como diferenças culturais, sociais e até mesmo o humor.

Analisando fisiologicamente, os seres humanos são animais homeotérmicos, logo tendem a manter a temperatura interna constante em qualquer ambiente em que se encontrem. Portanto, um ambiente que proporciona conforto térmico é um ambiente que permite manter a temperatura interna constante sem grandes esforços, de forma a alcançar o equilíbrio térmico com o meio [13].

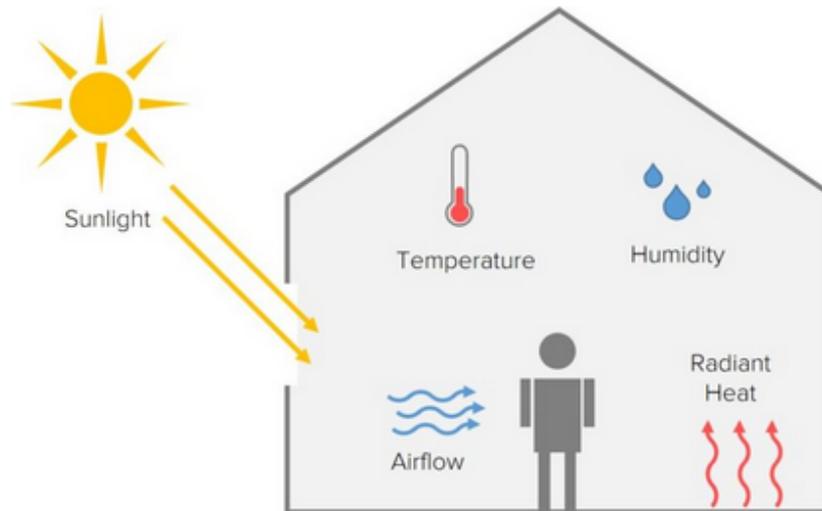


Figura 2.1: Representação dos fatores externos que influenciam o conforto térmico. Fonte: <https://ggbec.co.uk/quick-guide-overheating-thermal-comfort>

2.2 Termorregulação humana

De acordo com Lambert, podemos considerar o corpo humano como uma “máquina térmica” que dispõe de um mecanismo termorregulador que controla as variações térmicas do organismo. Sendo o organismo humano homeotérmico, o mecanismo termorregulador cria condições para que isso ocorra. Entende-se por “máquina térmica” aquela que necessita de certa quantidade de calor para seu funcionamento, de modo que esse funcionamento é a condição na qual o mesmo se encontra para que esteja apto a desempenhar suas atividades necessárias [1].

Para ter condições de desempenhar bem as atividades, nosso organismo necessita do calor, que é oriundo do metabolismo dos alimentos ingeridos. Uma parte desse calor gerado é necessário para o funcionamento fisiológico do organismo e a outra parte é gerada devido ao desempenho das atividades externas, sendo que esse calor deve ser dissipado para que não haja um superaquecimento do corpo [1].

Para que uma pessoa esteja em estado de conforto térmico no desempenho das atividades, admitem-se pequenas oscilações nessa temperatura interna, sendo que em situações mais extremas, admitem-se variações um pouco maior para se evitarem os perigos do estresse térmico [1].

Desta maneira podemos dizer que as atividades desempenhadas pelo ser humano geram calor ao corpo, o qual deve ser dissipado ao ambiente a fim de que não se ocasiona um aumento exagerado da temperatura interna, e que se mantenha o equilíbrio térmico do corpo. Essa dissipação se dá através de mecanismos de trocas térmicas, que podem ser observados na figura 2.2.

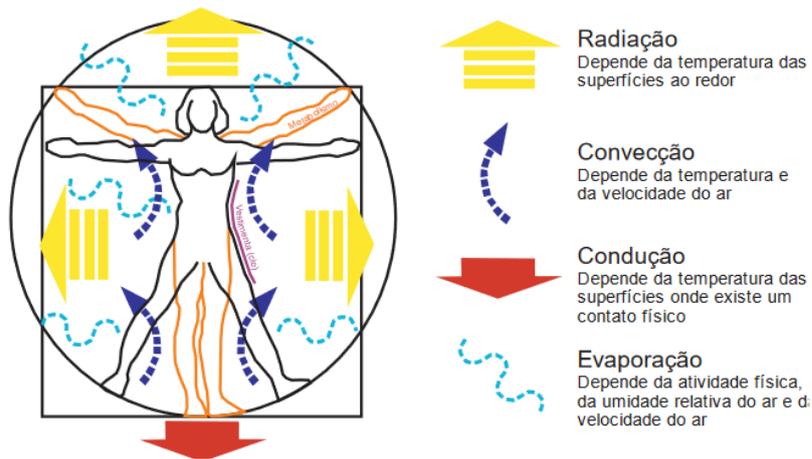


Figura 2.2: Representação da fisiologia humana levando em conta as trocas térmicas. [1]

2.3 PVM: Índice de conforto térmico

Tendo base em um estudo realizado com 1296 estudantes, Fanger propôs índices para avaliar o conforto térmico em um ambiente [14]. São eles:

- PMV: (*Predicted Mean Vote*)
- PPD (*Predicted Percentage of Dissatisfied*)

O índice PMV varia de -3 a +3, de modo que o valor máximo corresponde a muito quente e o valor mínimo muito frio. O valor equidistante do máximo e mínimo, ou seja, o zero, corresponde a melhor condição de conforto [15]. A equação abaixo expressa o cálculo do PMV:

$$PMV = 0.303 * (e^{-0.036 * M} + 0.028) * L \quad (2.1)$$

Nesta equação, o M representa a taxa de produção metabólica e o L representa a carga térmica atuante sobre o corpo do indivíduo. Através do PMV é possível realizar o cálculo do índice PPD, que denota a quantidade estimada de pessoas insatisfeitas termicamente com o ambiente. Ele se baseia na porcentagem de um grande grupo de pessoas que gostariam que o ambiente estivesse mais quente ou mais frio. A Equação abaixo expressa o cálculo do PPD:

$$PPD = 100 - 95 * e^{(-0.03353 * PMV^4 - 0.2179 * PMV^2)} \quad (2.2)$$

A figura 2.3 mostra a porcentagem de pessoas insatisfeitas (PPD) em função do Voto Médio Predito (PMV) [1].

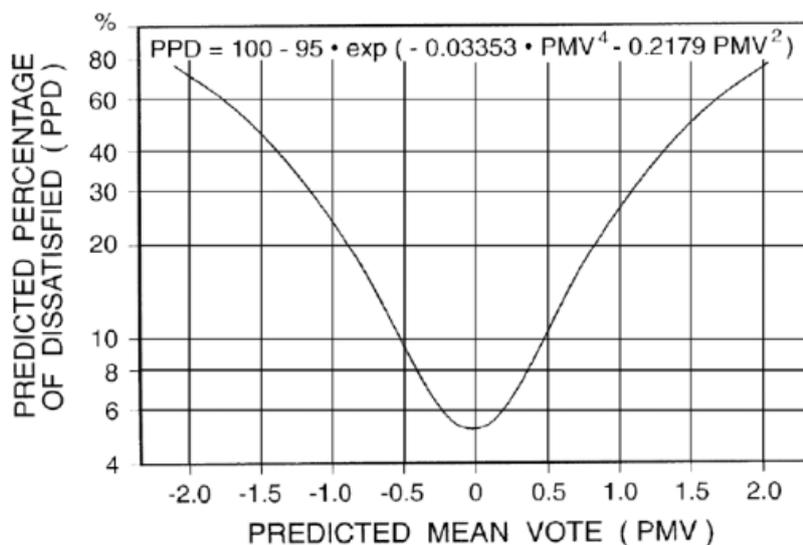


Figura 2.3: PPD em função do PPD [1].

Apesar de não considerar diferenças geográficas, a tabela 2.1 proposta pela ASHRAE, que é uma das normas mais aceitas no meio acadêmico [16], apresenta temperaturas recomendáveis para o conforto térmico:

	Temperatura Operacional	Intervalo aceitável
Verão	22°C	20-23°C
Inverno	24.5°C	23-26°C

Tabela 2.1: Padrão ASHRAE

Portanto, para a realização deste trabalho foi utilizado como base a temperatura operacional e seu intervalo aceitável recomendado pela norma ASHRAE.

2.4 Carga Térmica

De acordo com Creder, carga térmica é a quantidade de calor latente e sensível, geralmente expressa em BTU/h ou kcal/h, que deve ser retirada ou colocada no ambiente de modo a proporcionar as condições de conforto desejadas [17].

Calor sensível é a quantidade de calor que deve ser acrescentada ou retirada de um ambiente devido à diferença de temperatura entre o exterior e o interior, de modo a fornecer as condições de conforto desejadas. Esse calor é introduzido no recinto de diversas maneiras: por condução, pelo Sol diretamente, pelas pessoas, pela iluminação, pelo ar exterior e outros [17].

Já o calor latente é a quantidade de calor que se acrescenta ou retira de um corpo, causando a sua mudança de estado, sem mudar a temperatura. Por exemplo, é o calor absorvido que provoca a evaporação da água ou outros líquidos [17]. O calor emitido por seres humanos variam de acordo com diferentes fatores, como condições fisiológicas e atividade sendo exercida. Por

exemplo, geralmente homens emitem mais calor do que mulheres por causa de um maior trabalho muscular e metabolismo.

Neste trabalho foi levado em conta o gênero dos usuários nos ambientes para se criar uma estimativa de carga térmica.

2.5 RFID

RFID é um acrônimo para *Radio Frequency IDentification*, que significa identificação por rádio frequência. Essa tecnologia tem suas raízes na segunda guerra mundial, quando a utilizaram para a detecção de aproximação de alvos a muitos quilômetros de distância.

Esta tecnologia funciona por meio da transmissão de um sinal (ondas de rádio) entre uma antena, uma leitora e uma tag, que possui um circuito com uma informação a ser transmitida. Primeiramente a antena emite um sinal que se propaga no ar até chegar na tag, que por sua vez tem o circuito interno energizado pelo sinal. Isso permite que a tag emita um sinal de volta para a antena.

Ao receber o sinal de volta, a antena transmite as ondas de rádio do RFID para a leitora. Nesta última etapa o sinal é convertido em informação digital, que pode ser lido por um computador.

Essa tecnologia pode ser utilizada de diversas formas, como por exemplo a identificação e rastreamento de produtos, embalagens, animais e pessoas.

Neste trabalho foi utilizado RFID para o rastreamento de pessoas dentro de diferentes ambientes, de modo a monitorar esses ambientes e as transições dos usuários entre os mesmos.

Na figura 2.4 está ilustrado o funcionamento do RFID.

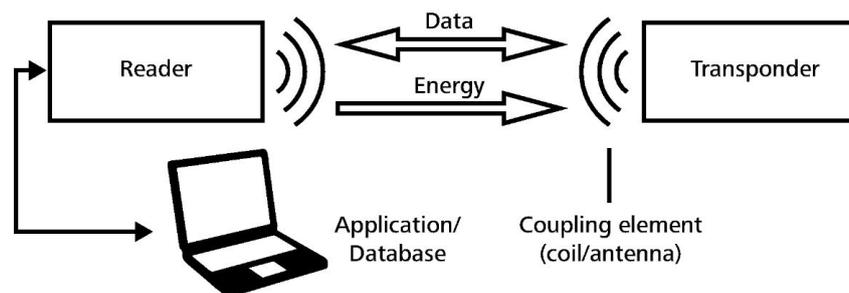


Figura 2.4: Exemplo de sistema RFID. Fonte: <https://www.hellermannntyton.com.br/>

2.6 Antenas

A antena é o equipamento que possui a função de enviar sinais de rádio e receber as informações emitidas pelas tags, podendo fazer isso a longa ou curta distância, dependendo do seu modelo, fabricante e aplicação.

Para funcionar corretamente, este componente precisa estar na mesma frequência da tag da qual vai ler as informações. A tag deve estar dentro do raio de transmissão da antena, e é desejado que não haja muitos obstáculos entre eles, pois isso pode causar interferência, impedindo a propagação da onda.

2.7 Tags

As tags, também conhecidas como etiquetas ou transpônderes, são dispositivos que possuem circuito interno que contém a informação a ser transmitida para a antena. Elas podem ser classificadas de duas formas:

- Passiva: utilizam o sinal emitido pela antena para energizar o circuito interno e transmitir uma informação. Geralmente estas informações estão gravadas de forma permanente no circuito.
- Ativa: possuem uma bateria própria como forma de energizar o circuito interno. Isso permite que a própria tag inicie comunicação com a leitora e que o sinal seja emitido a uma distância maior.

De uma forma mais básica, uma tag é composta por antena e circuito integrado fixados em um substrato, como pode-se ver na figura 2.5.



Figura 2.5: Componentes básicos de uma tag. Fonte: <https://rfidmoura.wordpress.com/>

2.8 EPC

O EPC, ou Código Eletrônico do Produto (*Electronic Product Code*), é utilizado como padrão de identificação único para cada tag. Portanto, é possível diferenciar cada tag por meio do seu EPC.

O padrão EPC define um esquema de numeração para as tags de tal forma que o número gerado é único, proporcionando uma exclusividade a cada tag.

Essa diferenciação é importante porque geralmente o objetivo do uso da RFID é a identificação de determinados produtos dentro de uma área específica. Para isso é essencial conseguir diferenciar um produto do outro.

Neste trabalho, cada usuário possui uma tag, proporcionando a individualização dos usuários dentro dos ambientes, permitindo uma análise de dados e ação personalizada.

2.9 Leitoras

A leitora possui a função de criar sinais de radiofrequência, transmiti-los através de uma antena, receber o sinal de resposta da tag pela antena e decodificar este sinal. Após isso, esses dados podem ser transmitidos para um computador para que seja possível fazer a análise desses dados.

A comunicação entre a leitora e o computador depende do modelo e fabricante. Geralmente essa comunicação pode ser feita utilizando Ethernet, Wi-fi, USB e RS-232, além de outros.

2.10 Middleware

O *middleware* é um programa de computador que faz a mediação entre outros softwares. Ele é utilizado para mover informações entre programas, ocultando do programador diferenças de protocolos de comunicação, plataformas e dependências do sistema operacional. [18]

O *middleware* de RFID é um serviço que se comunica e controla as leitoras de RFID a fim de recolher dados, que serão filtrados, agregados e consumidos por uma aplicação.

Dessa forma, após o *middleware* disponibilizar esses dados, diferentes aplicações podem utilizá-los, como por exemplo aplicativos baseados no EPC, sistemas de informação, ERPs (Planejamento de Recursos Empresariais) e outros aplicativos. Um exemplo de aplicação de RFID com *middleware* está ilustrado na figura 2.6.

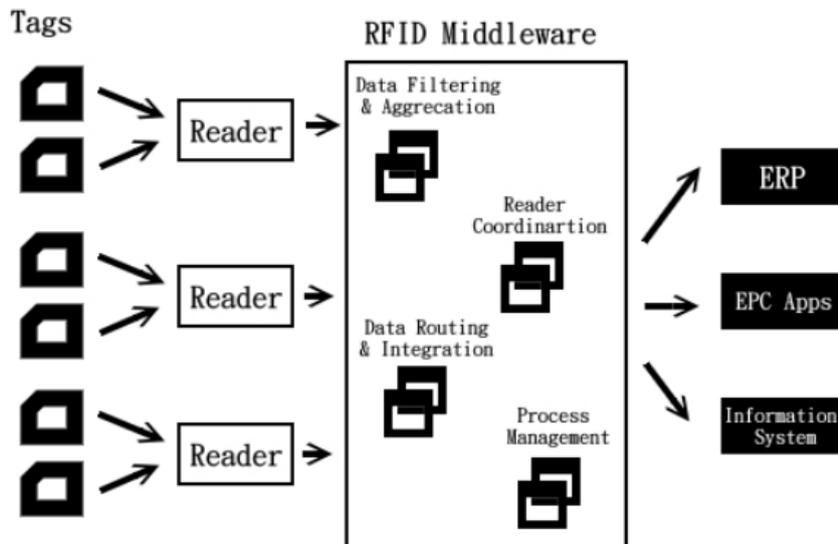


Figura 2.6: Exemplo de sistema de RFID com middleware [2].

2.11 RSSI

O RSSI, do inglês *Received Signal Strenght Indication*, é o indicador de intensidade do sinal recebido. Ou seja, o RSSI é uma medida da potência recebida do sinal retornado de uma tag RFID quando esta responde a um sinal enviado pela leitora.

Quando uma leitora reporta o valor RSSI de uma tag, ela está efetivamente reportando o nível de potência do sinal de resposta retro espalhado da tag no que se refere ao nível de potência do sinal de transmissão inicial da leitora. Este nível de potência é normalmente reportado em decibéis por miliwatt ou dBm [19].

Em aplicações RFID, os valores RSSI são usados para fornecer uma ideia geral de quão bem uma tag específica está respondendo em uma zona de leitura. Ao testar diferentes tags, o RSSI pode ser uma boa indicação de quão bem cada tag responderá em um ambiente semelhante. No entanto, os valores RSSI não fornecem uma medição totalmente precisa da distância da tag para a antena em um sistema RFID passivo. Isso se deve a muitos fatores ambientais que podem afetar o alcance de leitura em um sistema passivo [19].

Neste trabalho, foi utilizado o valor RSSI para fazer a detecção dos usuários durante as transições entre os diferentes ambientes. Através de testes, foi determinado um valor RSSI que correspondia aproximadamente a distância de cruzamento de uma antena. Caso a leitora reportasse um valor RSSI maior ou igual a este, significava que a tag estava passando na frente da antena.

2.12 Efeito Doppler

Efeito Doppler é um fenômeno físico que pode ser observado ao analisar as ondas emitidas ou refletidas por um objeto qualquer que esteja se movimentando em relação a um observador.

Ao se aproximar do observador, a frequência percebida por este em relação às ondas emitidas pelo objeto é maior, pois a cada vez que o objeto emite a onda ele se encontra em uma posição mais próxima. E ao se afastar, a frequência percebida pelo observador é menor porque a cada vez que o objeto emite a onda ele se encontra em uma posição mais distante.

Sabendo disso, em tese é possível determinar se uma tag lida pela leitora está se aproximando ou se afastando de um local.

2.13 Modos de leitura da Impinj Speedway Revolution R420

As leitoras Impinj Speedway Revolution R420 possuem 11 modos de leitura, que controlam a modulação utilizada para a transmissão de dados. A modulação de cada modo define, por sua vez, propriedades da leitora como taxa de leitura, área de leitura, sensibilidade e resistência a interferências.

As modulações (representadas pela letra M) indicam a quantidade de ciclos utilizados para encriptar cada bit de informação do sinal emitido entre as tags e leitoras. Assim, se $M = 2$, então para cada bit há dois ciclos de onda, se $M = 3$ então há 3 ciclos de onda para cada bit, e assim por diante. Essa relação pode ser melhor visualizada pela figura 2.7:

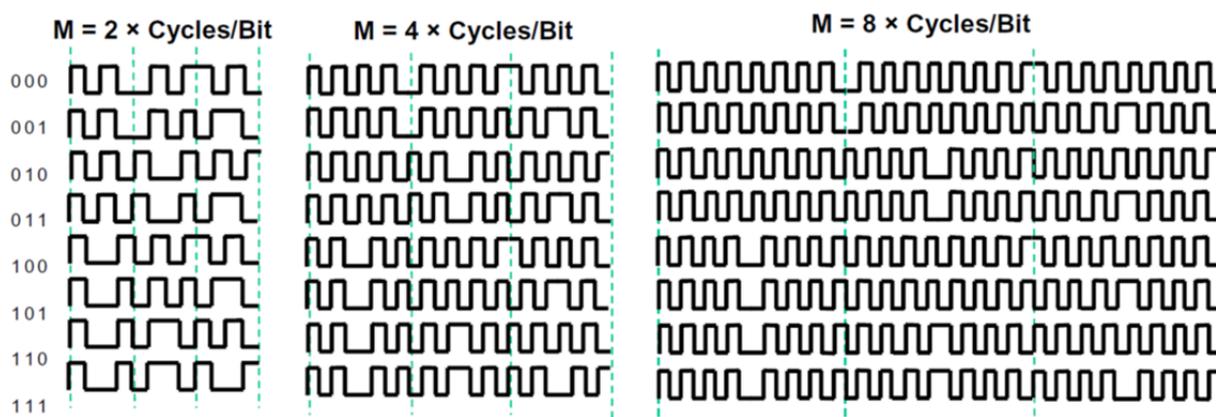


Figura 2.7: Fonte: Página do site da Impinj sobre modos de leitura

A modulação utilizada interfere diretamente na taxa de transmissão de dados. Quanto maior a modulação, maiores os ciclos necessários para transmitir cada bit e portanto será preciso mais tempo de processamento. Logo, a taxa de transmissão de dados é menor. A vantagem deste modo de operação é uma maior resistência a interferência à outras leitoras que estejam no mesmo ambiente ou outros dispositivos que por acaso trabalhem com sinais na mesma faixa de frequência,

resultando em uma maior confiabilidade dos dados.

Por outro lado, quanto menor a modulação utilizada maior será a taxa de leitura. Essa configuração é interessante para ambientes mais dinâmicos, como por exemplo competições de corrida de carros. Nesses casos as tags ficam nos carros, enquanto as leitoras ficam na linha de chegada. Se pelo menos dois carros cruzam a linha de chegada numa pequena faixa de tempo, é necessário que as leitoras possuam uma alta taxa de leitura para conseguir diferenciar o instante em que cada carro passou pela leitora.

Portanto, maiores modulações não implicam necessariamente que o resultado será melhor. Tudo depende do objetivo da aplicação das leitoras. No caso deste projeto, queremos monitorar a presença e transição de tags entre ambientes adjacentes. Para escolher o melhor modo de leitura a ser utilizada, precisamos responder primeiro duas perguntas: em que contexto e ambiente as leitoras serão utilizadas? Quais as características de cada modo de leitura?

Como explicado anteriormente, as leitoras serão instaladas em um ambiente onde haverá deslocamento de pessoas que carregam consigo as tags. O objetivo é o monitoramento da presença e detecção de passagem das tags entre salas adjacentes. Portanto, a taxa de leitura não precisa ser muito alta, pois a velocidade de deslocamento das tags será sempre igual à velocidade de deslocamento das pessoas, que provavelmente estarão andando em baixa velocidade. No entanto, precisa ser alta o suficiente para detectar a passagem de uma pessoa por uma porta à uma velocidade mínima.

Entendido o contexto em que o equipamento será aplicado, é preciso compreender as características de cada modo de leitura. Elas são:

- Mode 0: possui a maior taxa de leitura, no entanto é a mais suscetível a interferências;
- Mode 1: utiliza $M = 2$;
- Mode 2: utiliza $M = 4$;
- Mode 3: utiliza $M = 8$. Modo mais resistente a interferências;
- Mode 4: utiliza $M = 4$. Possui alta taxa de leitura e funciona com múltiplas leitoras no mesmo ambiente;
- Mode 5: utiliza $M = 4$. Possui alta taxa de leitura e funciona com múltiplas leitoras no mesmo ambiente;
- Mode 1000: utiliza modulações variadas. Analisa as ondas de radiofrequência do ambiente e depois escolhe automaticamente um modo de leitura entre 1 e 5;
- Mode 1002: utiliza modulações variadas. Combina os modos com alta taxa de leitura e os modos com baixa taxa de leitura. Maximiza o número de leituras de tags únicas;
- Mode 1003: utiliza modulações variadas. Combina apenas os modos com alta taxa de leitura. Possui alta taxa de leitura e funciona com múltiplas leitoras no mesmo ambiente;

- Mode 1004: utiliza modulações variadas. Combina os modos de leitura 2 e 3;
- Mode 1005: utiliza modulações variadas. Uso não recomendado;

Foram feitos testes utilizando cada modo de leitura descrito acima. Para definir o melhor modo de leitura, foram considerados critérios como: taxa de leitura, área de leitura e sensibilidade.

2.14 Relação entre área de leitura e sensibilidade

Foi feito um teste pela Impinj para observar a relação entre a sensibilidade, taxa de leitura e a área de leitura das leitoras. A documentação deste teste pode ser encontrada no site da Impinj. Foram utilizados 20 tags posicionados à frente de uma antena e o teste foi repetido para cada modo de leitura. O resultado é ilustrado pela figura 2.8.

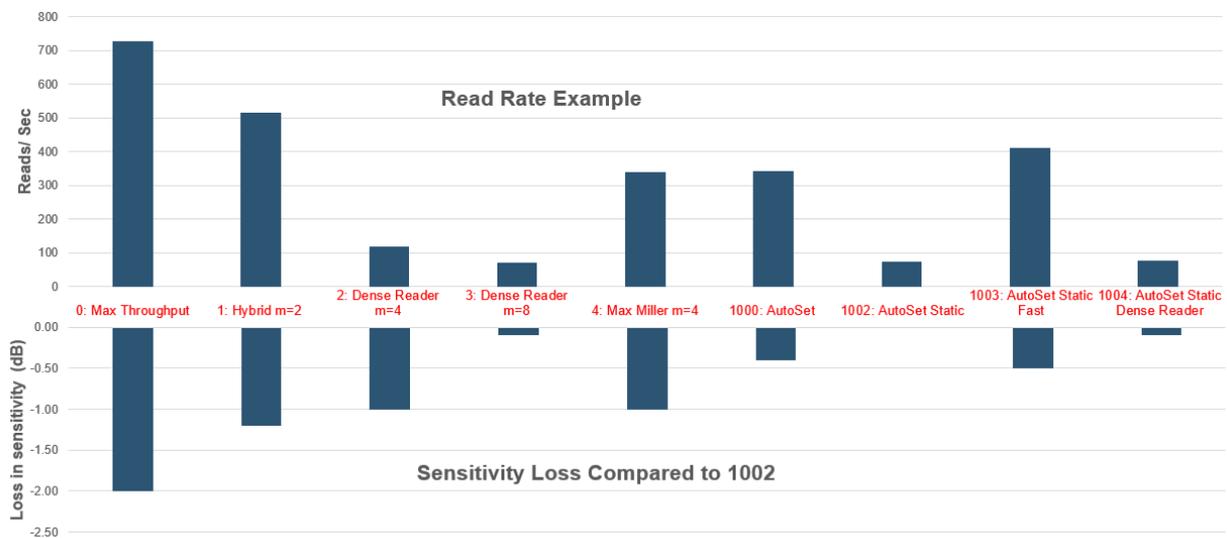
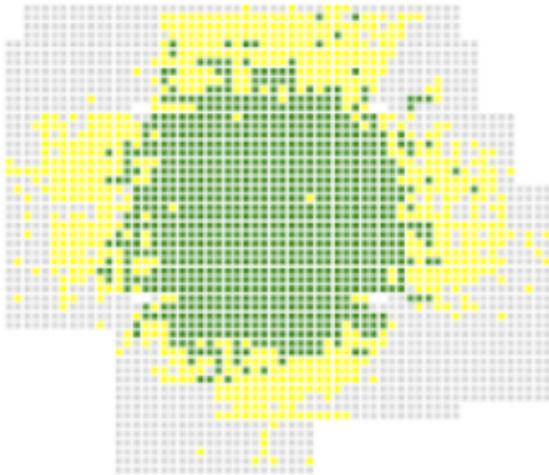


Figura 2.8: Fonte: Página do site da Impinj sobre modos de leitura

O objetivo do teste era identificar qual modo de leitura conseguia detectar o maior número de tags únicas e altamente concentradas, ou seja, muito próximas. O resultado mostra uma diferença de 2 dB entre o modo de leitura mais sensível (modo 1002) e o modo menos sensível (modo 0). Esses 2 dB representam a diferença de intensidade do sinal lido pelas leitoras.

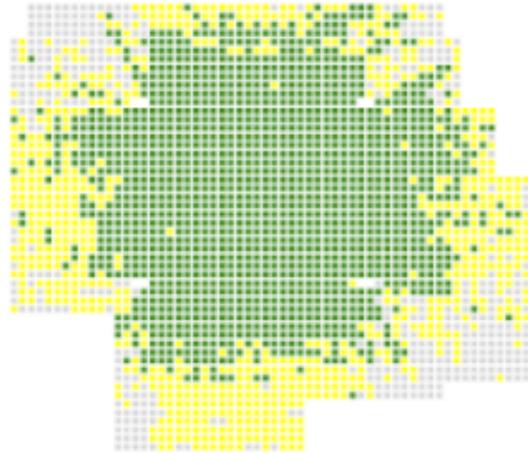
A diferença de intensidade do sinal lido implica em uma diferença de área de leitura coberta pela antena. De fato, foi observado uma diferença de 26% entre as áreas de leitura do modo 0 e do modo 1002. Essa diferença pode ser observada pela figura 2.9.

Mode 0: Max Throughput



Sq m : 144
Sq ft : 1545

Mode 1002: AutoSet Dense Reader Deep Scan



Sq m : 205
Sq ft : 2205

Figura 2.9: Fonte: Página do site da Impinj sobre modos de leitura

Pela figura 2.9 pode-se observar que quanto menos sensível é o modo de leitura (modo 0), menor será a área de leitura coberta pela antena. Quanto mais sensível é o modo de leitura (modo 1002), maior será a área de leitura coberta pela antena.

2.15 Trade-off entre taxa de leitura e resistência à interferências

Relembrando o que foi explicado na seção de Modos de leitura da Impinj Speedway Revolution R420, há uma relação entre taxa de leituras e quantidade de modulações utilizadas pelas leitoras para encriptar os bits. Quanto maior a quantidade de modulações, maior será o tempo de processamento. Portanto, menor será a taxa de leitura.

Por outro lado, a quantidade de modulações também está relacionado à resistência da leitura à interferências. Quanto maior a quantidade de modulações, maior será a resistência. E quanto menor for a quantidade de modulações, menor será a resistência.

Juntando as duas ideias, pode-se relacionar a taxa de leitura com a resistência à interferências. Uma alta taxa de leitura implica em uma menor quantidade de modulações. Isso causa uma menor resistência à interferências. Portanto, um modo de leitura com estas características consegue detectar passagens mais rápidas das tags, no entanto estão mais suscetíveis a erros. Uma baixa

taxa de leitura resulta numa situação inversa. Neste caso, um modo de leitura com baixa taxa de leitura e alta resistência à interferências não conseguiria detectar a passagens de tags caso elas estivessem se deslocando com uma velocidade muito alta. No entanto, os dados de leituras são mais confiáveis.

Dado a aplicação deste projeto e as condições onde o trabalho foi executado, não é necessário uma taxa de leitura muito elevada. Por se tratar de um ambiente fechado e com possíveis interferências de outros equipamentos, a resistência a interferência é a característica mais desejável para a execução deste projeto.

Capítulo 3

Materiais e Ferramentas

3.1 Visual Studio

O ambiente de desenvolvimento integrado do Visual Studio é um painel de inicialização criativo que pode ser usado para se editar, depurar e compilar o código e, em seguida, publicar um aplicativo. Um IDE (ambiente de desenvolvimento integrado) é um programa repleto de recursos que pode ser usado por muitos aspectos do desenvolvimento de software. Além do editor e do depurador padrão fornecidos pela maioria dos IDEs, o Visual Studio inclui compiladores, ferramentas de preenchimento de código, designers gráficos e muitos outros recursos para facilitar o processo de desenvolvimento de software [20].

Através do Visual Studio foi utilizado o kit de desenvolvimento Octane SDK disponibilizado pela Impinj. Este trabalho utilizou a biblioteca Octane SDK, que permite a programação com a linguagem em C#, para fazer a configuração das leitoras e antenas de forma a realizar análises em cima das leituras das tags.

3.2 Windows Forms

Windows Forms é um *framework* de interface do usuário para a construção de aplicativos de desktop do Windows. Ele fornece uma das maneiras mais produtivas de criar aplicativos de área de trabalho com base no designer visual fornecido no Visual Studio. Funcionalidades como o posicionamento de arrastar e soltar controles visuais facilita a criação de aplicativos de desktop [21].

Com o Windows Forms, é possível desenvolver aplicativos graficamente ricos que são fáceis de implantar, atualizar e trabalhar offline ou conectado à Internet. Os aplicativos Windows Forms podem acessar o hardware local e o sistema de arquivos do computador em que o aplicativo está sendo executado [21].

No Windows Forms, um formulário é uma superfície visual na qual são exibidos informações ao usuário. Normalmente, é criado aplicativos do Windows Forms adicionando controles a formulários e desenvolvendo respostas às ações do usuário, como cliques do mouse ou pressionamentos de teclas. Um controle é um elemento discreto de interface de usuário que exibe dados ou aceita entrada de dados [21].

Neste trabalho o Windows Forms foi utilizado para criar um aplicativo de interface gráfica para visualização das transições de pessoas entre os ambiente, assim como a quantidade de pessoas por ambiente e outras informações de interesse.

3.3 Leitoras Impinj Speedway Revolution r420



Figura 3.1: Ilustração da leitora Impinj Speedway Revolution R420.

A figura 3.1 representa a leitora Impinj Speedway Revolution R420. Este é um equipamento usado para detectar objetos que carregam uma tag usando radiofrequência. Ele foi feito para manter uma alta taxa de leitura e detecção de tags passivas mesmo em ambientes com interferência e ruídos de radiofrequência.

O Impinj Speedway Revolution R420 é programável por meio da biblioteca Octane SDK, uma biblioteca de desenvolvimento feito para suportar aplicações em C# e Java para leitores da Impinj.

Essa leitora possui a função de autopilot e autotset, que automaticamente ajustam as configurações da leitora de acordo com o ambiente.

Em cada leitora podem ser acopladas até 4 antenas e a alimentação pode ser feita através de uma fonte ou através de POE (*Power Over Ethernet*). Neste projeto foram utilizadas duas antenas por leitora.

A ligação entre a leitora, o switch e o computador deve ser feito de acordo com o que é mostrado na figura 3.2.



Figura 3.2: Exemplo de rede configuração de rede utilizando cabo ethernet e switch de rede. Imagem retirada de [3].

3.4 Antena Impinj Threshold



Figura 3.3: Imagem da antena Impinj Threshold. Imagem retirada de [4]

As antenas Impinj Threshold (ilustradas na figura 3.3) são utilizadas para a detecção de passagem de tags e são ideais para detectar a passagem destas por algum determinado limite, neste caso as divisões entre os ambientes. Elas detectam as tags assim que entram em sua zona de leitura e foram projetadas para detectar sinais intensos em regiões específicas.

Neste trabalho, as antenas ficam posicionadas nas portas e passagens de cada ambiente monitorado e são utilizadas duas dessas antenas por leitora. Elas devem ser conectadas na parte traseira da leitora, nos locais indicados na figura 3.4.

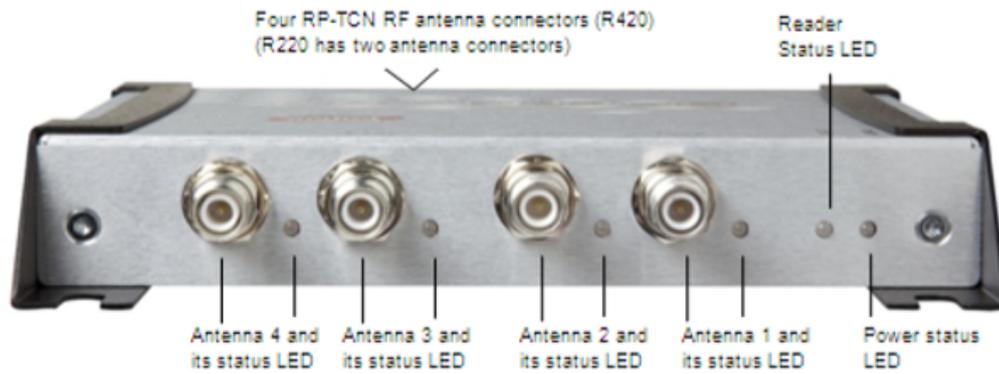


Figura 3.4: Portas da antena no Reader Speedway R420. Imagem retirada de [3].

3.5 Tag Rfid Uhf Adesiva 915 - 868 Mhz Higgs3 Passiva



Figura 3.5: Tags RFID passivas utilizadas no projeto.

As tags utilizadas no projeto são tags passivas, ilustradas na figura 3.5. Foram utilizadas 10 tags para simular um trânsito de 10 pessoas entre os ambientes monitorados.

Essas tags são adesivas, possuem um tamanho de 96x23mm, 512 bits de memória e operam em uma faixa de frequência entre 860MHz e 960MHZ.

3.6 ItemTest

Após a montagem das leitoras e antenas, foi utilizado o software ItemTest, da Impinj, para realizar testes iniciais de leitura para verificar o funcionamento correto dos equipamentos.

Este software tem a capacidade de fazer a leitora funcionar sem a necessidade de programação, mostrando informações de leitura como intensidade do sinal lido da tag, indicador de distância da tag, quantidade de leituras, número da antena que detectou o sinal da tag, etc.

Com o ItemTest é possível verificar o funcionamento de vários componentes. Isso facilita o processo de revisão dos componentes quando eles não estão funcionando corretamente.

A interface está ilustrada na figura 3.6.

EPC	Read #	ef-35	Average RSSI	Max RSSI	Min RSSI
55504D10000003000419093	30	30	-62.0	-61.5	-63
55504D10000003000381577	64	64	-57.6	-55.5	-61.5
00000000000000000000095	72	72	-63.4	-61.5	-67
E20064438B37E8B12B0FEFA2	23	23	-69.4	-69	-70
E20064438B30E9F12B0FD3A7	43	43	-65.9	-65	-67
E2009A609008AF000000656	28	28	-60.7	-60	-62
300833B2DD9014000000000	16	16	-67.1	-66.5	-67.5
E280116060002052A0E6D91	18	18	-68.3	-67.5	-69.5

Figura 3.6: Interface do software ItemTest. Imagem retirada de [5].

As linhas em branco da figura 3.6 indicam que as respectivas tags estão sendo lidas frequentemente, enquanto as linhas em vermelho indicam que as respectivas tags não foram lidas há um tempo.

Para configurar o ItemTest, basta seguir as instruções que estão em anexo.

3.7 Octane SDK

A biblioteca utilizada para programar as leitoras foi a Octane SDK. Ela é uma biblioteca de desenvolvimento que suporta aplicações em C# e Java para leitoras da Impinj.

A biblioteca Octane SDK facilita a programação das leitoras utilizadas, tornando mais rápido e fácil as configurações destas. A biblioteca possui documentação e vem com vários códigos exemplos para diferentes finalidades da leitora. Por causa disso, é possível focar no desenvolvimento da lógica de programação para o objetivo deste projeto, ao invés de se preocupar com a programação em baixo nível do hardware.

3.8 Roteador TP-Link

Para que haja troca de informação entre o computador e a leitora é preciso utilizar um roteador, que atuará como ponte para o tráfego de dados.

O roteador utilizado no projeto é chamado de **TP-Link TL-WR940N** e está ilustrado na figura 3.7.



Figura 3.7: Ilustração da parte frontal do roteador.

Na parte traseira do roteador há 5 portas para conectores RJ45, das quais foram utilizadas 4: uma para o computador e 3 para as leitoras. O local para conexão dos 4 conectores está destacado na figura 3.8.



Figura 3.8: Ilustração da parte traseira do roteador.

Além das conexões por cabos, nenhuma configuração por software é necessária.

3.9 Arduino

Arduino é um dispositivo criado em 2005 para que projetistas e estudantes amadores pudessem aprender a projetar circuitos e programar de forma mais acessível e fácil.

O arduino é composto por um microcontrolador Atmel, além de entradas e saídas que podem ser conectadas a outros sensores ou até a um computador. Ele é programado através da IDE do Arduino e a linguagem utilizada é baseada em C/C++.

Existem diversas versões do arduino, como por exemplo arduino uno, arduino mega, arduino nano e muitos outros. Cada um possui suas peculiaridades e a escolha de qual deve ser utilizado dependerá das especificações do trabalho que deve ser executado. Para este projeto o objetivo do arduino é se comunicar com o computador e ler as informações transmitidas por ele. Com base no que foi lido deve o arduino deve ativar ou desativar uma carga através de um relé. Por isso um arduino básico, como o uno, já cumpre com os requisitos e consegue lidar com o trabalho proposto.

Segue abaixo uma ilustração (figura 3.9) do Arduino Uno:

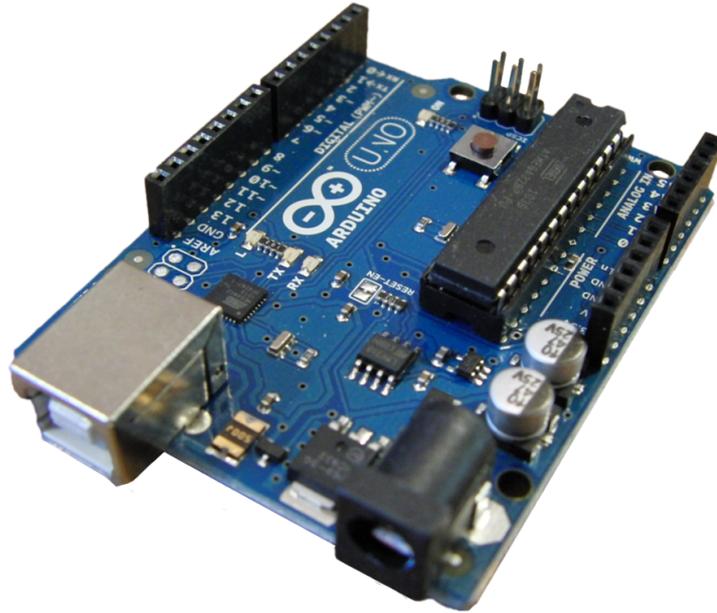


Figura 3.9: Ilustração do arduino Uno.

3.10 Relé

Um relé, também chamado de relê, é um interruptor eletromecânico que é usado para fazer o acionamento de uma carga e ao mesmo tempo isolar circuitos. No caso do projeto, o relé isola o circuito do microcontrolador do circuito da carga, pois cada um lida com tensões diferentes. Enquanto o arduino trabalha com tensões de até 5V, a carga trabalha com tensões de 220V.

O objetivo inicial deste trabalho era acionar um aparelho de ar condicionado, especificamente o que fica na sala do LARA, no prédio SG 11 da Universidade de Brasília. No entanto, devido à pandemia de COVID-19 que ocorreu durante o período de execução deste projeto não foi possível acessar a sala do LARA, e por isso não houve a possibilidade de trabalhar com o aparelho de ar condicionado.

Para simular o acionamento do aparelho de ar condicionado utilizamos três lâmpadas LED (uma para cada sala), pois elas se destacariam no vídeo que seria gravado mostrando o funcionamento do projeto.

Segue abaixo uma ilustração (figura 3.10) do relé utilizado:



Figura 3.10: Ilustração do relé.

3.11 Lâmpada LED

As lâmpadas LED utilizadas foram lâmpadas LED comuns, como a ilustrada na imagem abaixo:



Figura 3.11: Ilustração da lâmpada LED utilizada.

Essas lâmpadas (ilustradas na figura 3.11) foram utilizadas por causa da impossibilidade de acesso aos aparelhos de ar condicionado do LARA. Estes eram acionados por meio de um relé, que era controlado por um arduino uno. Para solucionar este problema substituímos os aparelhos de ar condicionado pelas lâmpadas LED, pois estas poderiam representar o acionamento de uma carga qualquer.

Capítulo 4

Experimentos e Resultados

4.1 Introdução

Para a tentativa de detecção de passagem de pessoas foram adotadas duas estratégias: uma usando as informações de efeito Doppler e outra usando as informações de RSSI, ambas providas pela leitora Speedway Revolution R420.

Em ambas as estratégias foi posicionado duas antenas em série a uma distância de 50 centímetros entre elas e então foi obtido as informações de leitura das tags.

Foram desenvolvidos programas utilizando o Visual Studio e o kit de desenvolvimento Octane SDK disponibilizado pela Impinj de forma a configurar as leitoras e antenas, obter dados de leitura das tags e realizar a análise de dados necessária para cada experimento.

A abordagem escolhida para a detecção de transição de um usuário para outro ambiente foi o rastreamento das tags por proximidade das antenas, de modo que quando o valor de leitura de uma antena ultrapassava um valor mínimo preestabelecido, era detectado a passagem de uma tag.

Para identificar a direção da passagem foi utilizado o tempo de pico do sinal lido, ou seja, o instante em que o valor lido ultrapassava o valor mínimo preestabelecido. Para cada antena há um pico do sinal em um tempo diferente, pois elas estão posicionadas em série na divisória entre dois ambientes. Desta forma, a tag sempre irá passar na frente de uma antena antes da outra. A comparação do tempo de pico entre as antenas nos fornece a informação do sentido de passagem da tag, identificando a transição do usuário de um ambiente para o outro.

4.2 Posicionamento das leitoras

Há muitos debates quanto a usar área de leitura de uma leitora como indicador de desempenho. No entanto, é extremamente complexo o cálculo desta área devido às muitas variáveis que afetam o sistema. Além do sinal transmitido entre tags e antenas, há também a interferência de outros equipamentos que podem estar na mesma sala e que também utilizam radiofrequência, há a interferência causada pelos múltiplos caminhos que um sinal percorre após ser emitido e existe

também a forma como a tag está orientada em relação à antena. Todos esses fenômenos dificultam o cálculo da área de leitura dos dispositivos.

Devido à dificuldade do cálculo da área de leitura, encontramos empiricamente a melhor disposição das leitoras e antenas para obter leituras mais acuradas e precisas das tags. A disposição está ilustrada na figura 4.1.

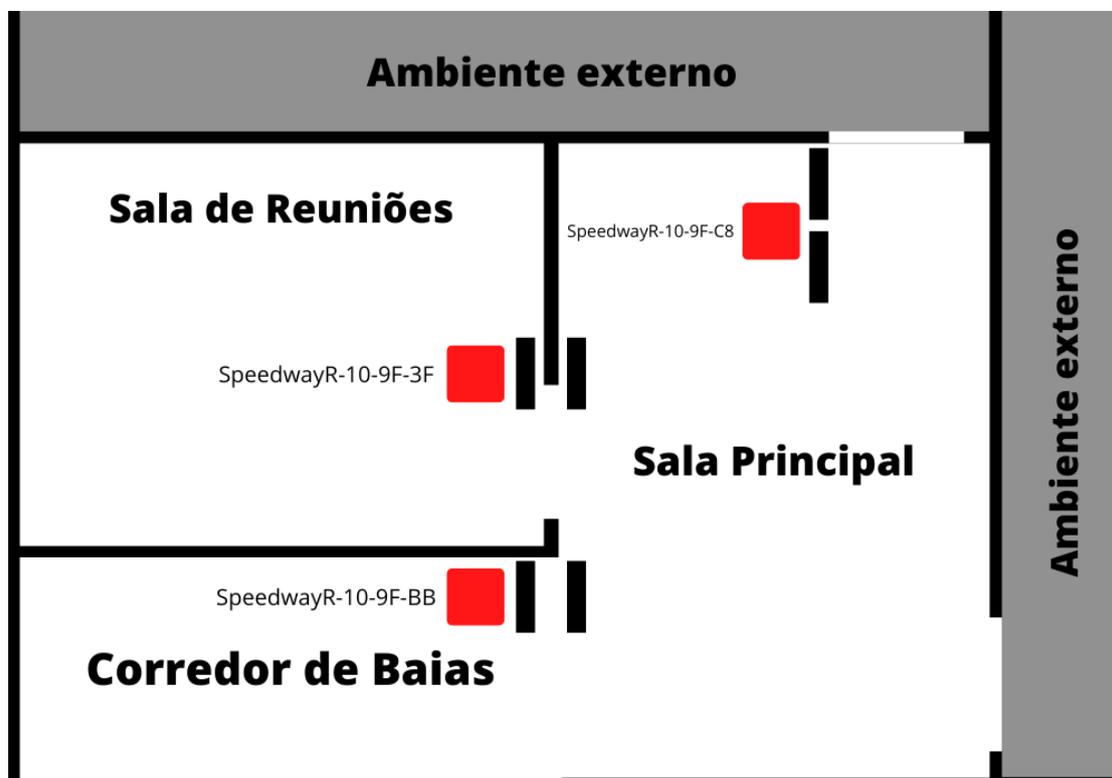


Figura 4.1: Disposição das leitoras e antenas no LARA.

Os elementos em vermelho e preto da figura são explicados na figura 4.2:

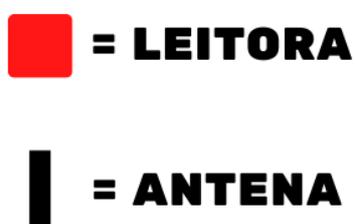


Figura 4.2: Legenda da figura 4.1

A figura 4.1 representa as salas do lado esquerdo do LARA, localizado no prédio SG-11 da Universidade de Brasília. Os ambientes da figura são:

- Sala de Reuniões: sala onde há a mesa, o projetor usado para fazer reuniões e um ar condicionado;
- Corredor de Baias: local onde fica a mesa com computadores e um ar condicionado;
- Sala Principal: ambiente ao lado da porta esquerda de acesso ao LARA;
- Ambiente externo: locais externos aos ambientes considerados até agora. Inclui o espaço do lado direito do LARA e a área externa desta;

Como pode ser visto na figura 4.1, as leitoras e antenas foram posicionadas sempre nas portas ou áreas de transição entre os ambientes. Isso garante que as transições das tags entre os ambientes sejam sempre detectadas, pois é necessário passar pelas portas onde estão localizadas as antenas para transitarem entre as salas.

Devido à quarentena provocada pela pandemia de COVID-19 que está ocorrendo no momento da realização deste projeto, foi impossibilitado a utilização do espaço do LARA. Por isso transportamos e instalamos o equipamento em uma casa vazia.

Escolhemos uma casa que possui três quartos que podem simular o espaço do LARA. Isso torna mais fácil uma possível replicação futura do projeto no SG-11.

Segue abaixo (figura 4.3) uma ilustração da planta da casa, seguindo a mesma legenda da figura 4.1:

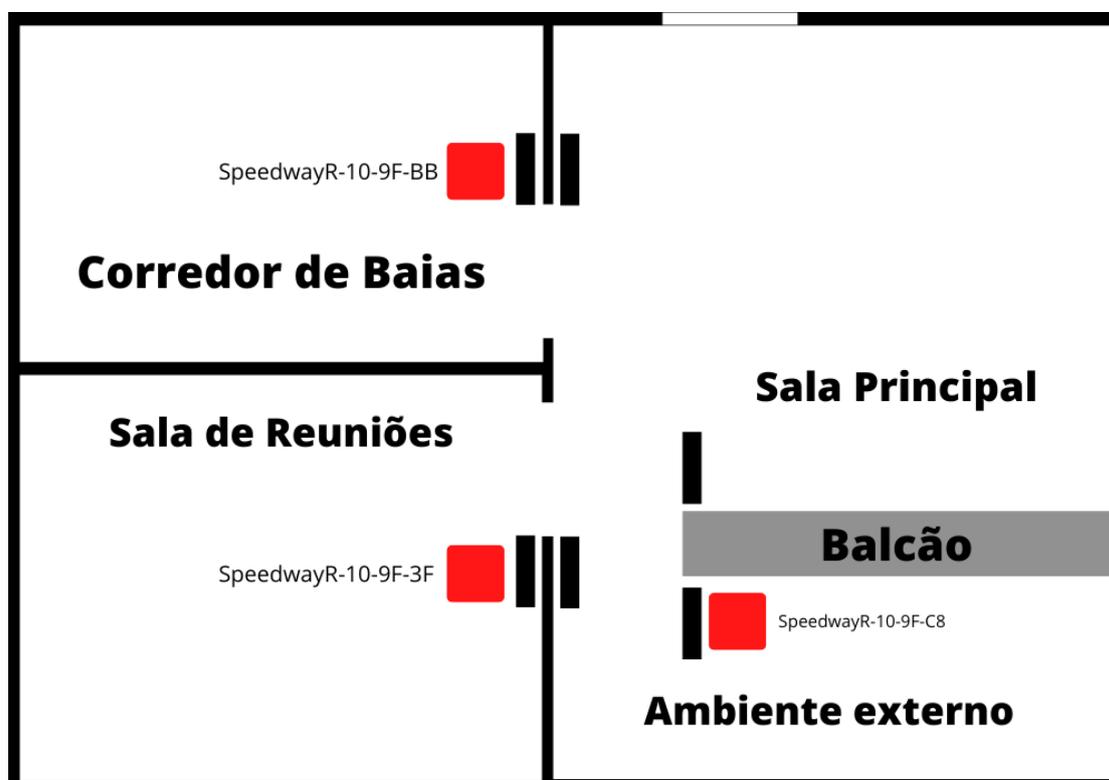


Figura 4.3: Planta da casa onde foi implementado o projeto.

Seguem abaixo imagens reais da casa (figuras 4.4, 4.5 e 4.6) onde foi implementado o projeto:



Figura 4.4: Imagem geral da casa.





Figura 4.6: Imagem da área externa.

4.3 Detecção de pessoas usando informações do efeito Doppler

Como dito anteriormente, uma das abordagens para a tentativa de detecção da transição de pessoas entre os ambientes foi a utilização da frequência doppler, que é um parâmetro que se pode obter diretamente da leitora da Impinj sem a realização de cálculos.

A medida da frequência Doppler pode ser obtida através um comando da biblioteca Octane SDK e o valor é uma estimativa feita a partir do cálculo do tempo que um sinal leva para chegar na tag e voltar à antena. A unidade da medida é em hertz.

Para a detecção da passagem de tags utilizando as leituras do Efeito Doppler consideramos 4 hz como o valor mínimo para considerar uma passagem. Foi observado que valores abaixo desse poderiam ser considerados como falso positivo. Também foi observado que valores muito acima de 4 hz são atingidas apenas se a tag passar pela leitora em uma velocidade muito alta, como por exemplo se a pessoa estivesse correndo.

A figura 4.7 ilustra um exemplo de leitura da frequência Doppler em relação ao tempo com uma passagem de uma tag. Quando o valor lido passa de 4 hz, foi considerado que alguma tag está passando em frente à antena.

A taxa de leitura do equipamento utilizado é de aproximadamente 10 leituras por segundo. Por isso haverá múltiplas leituras até a tag se afastar da antena após uma passagem, ou seja, até o valor lido voltar a ser menor que 4. Por causa disso, é bem provável que o valor lido oscile entre 4, causando erroneamente a impressão de diversas passagens. Por exemplo, pode-se observar no gráfico da figura 4.7 que o valor da frequência doppler lido oscilou algumas vezes entre 4 hz. Isso indicaria no sistema que ocorreu mais de uma transição.

Para contornar este problema foi definido um tempo mínimo de 700 mili-segundos após a detecção de uma passagem para considerar a detecção de outra passagem.

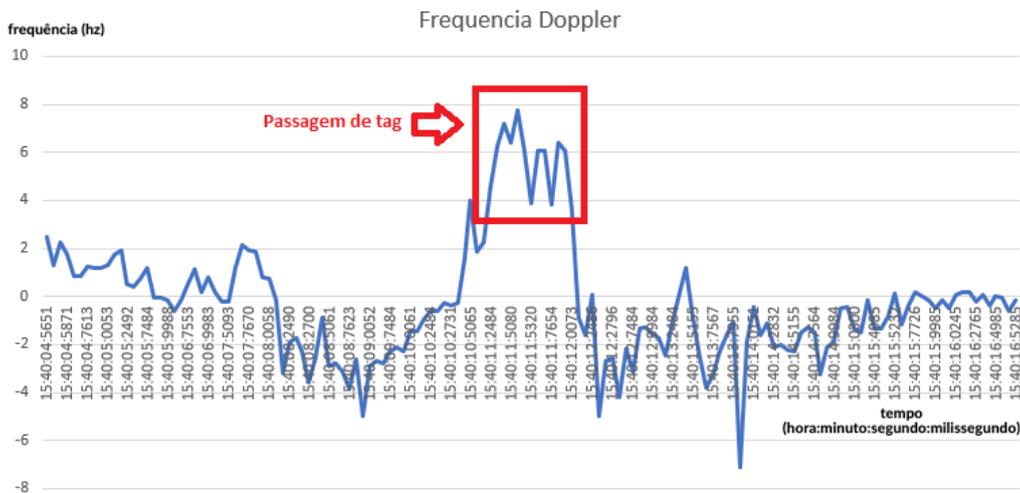


Figura 4.7: Exemplo de leitura de passagem de uma tag por uma antena.

4.3.1 Experimento utilizando valor mínimo de 4 hz de frequência doppler para considerar passagem

Após a validação de que é possível realizar a detecção de uma passagem da tag utilizando a frequência doppler, foram feitos alguns experimentos fazendo 10 passagens por duas antenas em série com a mesma tag. Os resultados obtidos para a antena 1 e para a antena 2 estão separados nas próximas subseções.

4.3.1.1 Dados do experimento para a antena 1

A figura 4.8 ilustra o gráfico da frequência doppler em função do tempo para a antena 1.

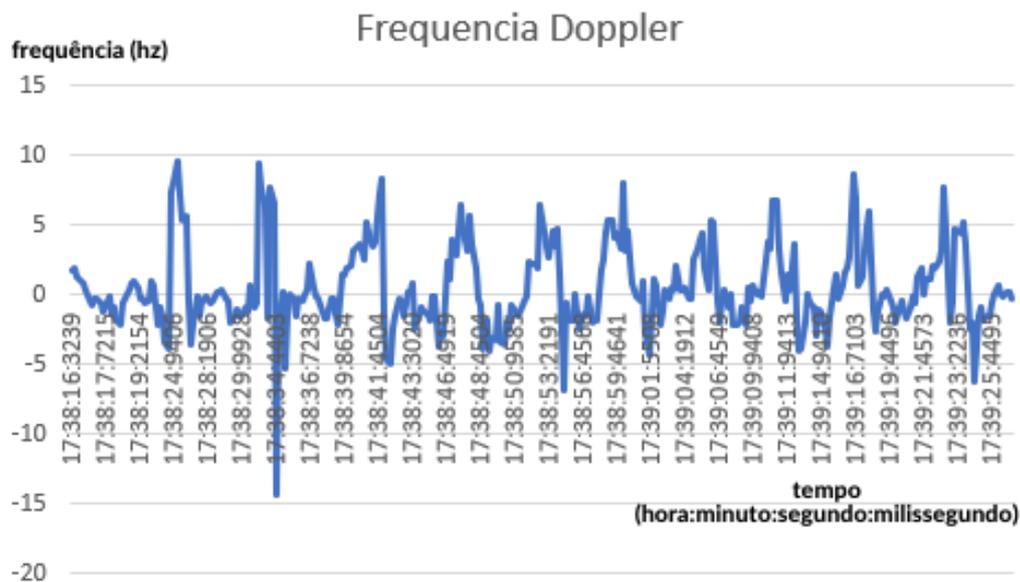


Figura 4.8: Leitura da frequência Doppler para a antena 1.

Observando esse gráfico, pode-se perceber que visualmente existem destacados 10 picos, representando um resultado otimista em relação às 10 passagens realizadas.

Entretanto, observando o gráfico da detecção de passagens ilustrado na figura 4.9, pode-se perceber que foram contabilizadas 13 transições. Isso ocorreu porque em algumas passagens da tag pela antena a oscilação do valor lido causou erroneamente a detecção de mais de uma passagem ao mesmo tempo.

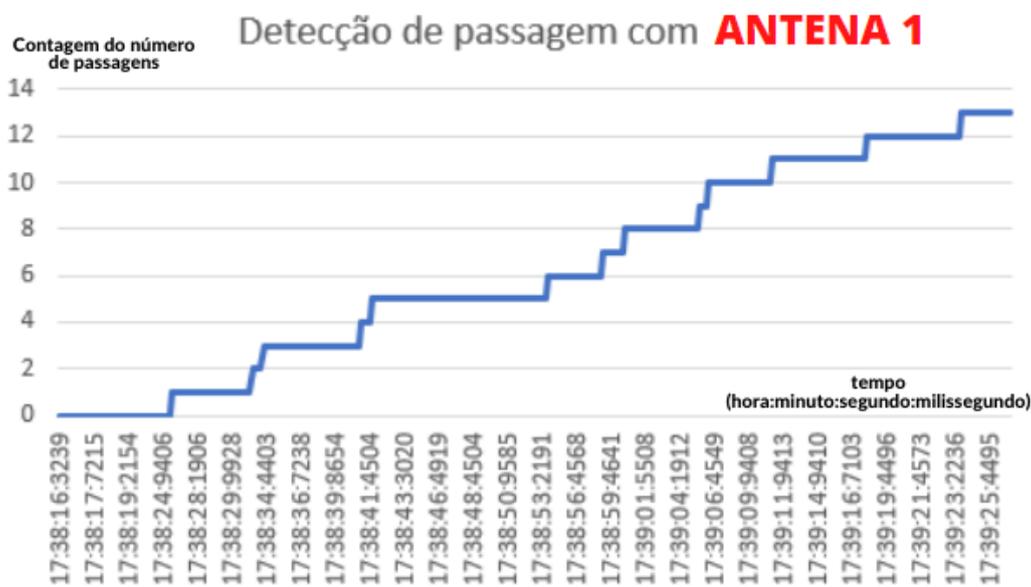


Figura 4.9: Contagem da detecção de passagens pela antena 1.

4.3.1.2 Dados do experimento para a antena 2

A figura 4.10 ilustra o gráfico da frequência doppler em função do tempo para a antena 2. Observando esse gráfico, pode-se perceber que as leituras obtidas não foram muito satisfatórias, não havendo uma curva de aproximação e afastamento muito bem definidas, como foi observado no resultado obtido pela antena 1.

Realizando o gráfico da contagem de passagens da tag pela antena 2, ilustrado na figura 4.11, percebe-se que foram contabilizados apenas 7 passagens. O valor de frequência doppler lido oscilou muito e em alguns momentos não atingiu o valor mínimo preestabelecido, provocando erros na contagem de passagens.

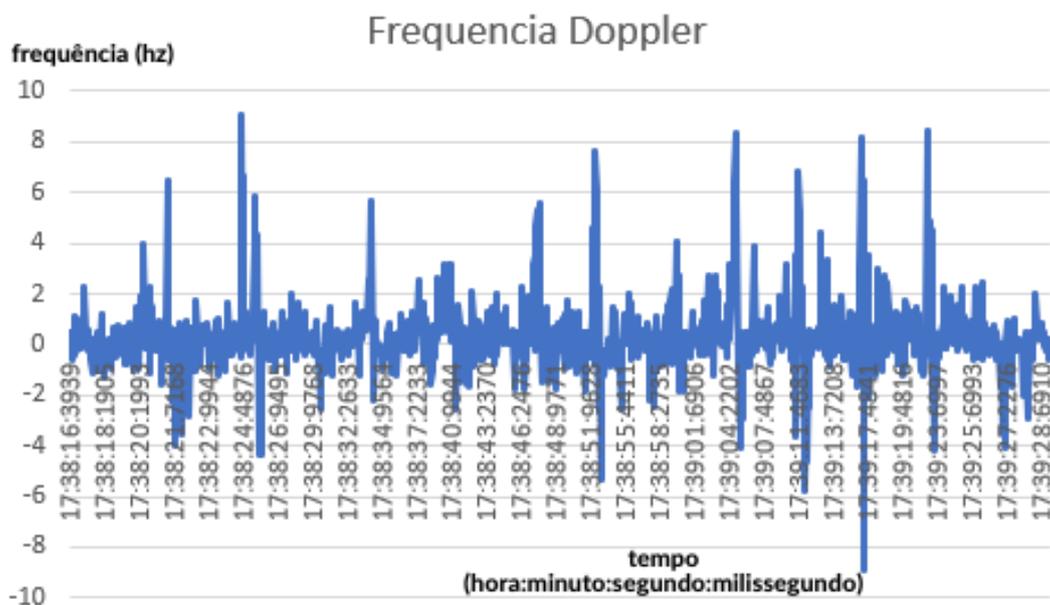


Figura 4.10: Leitura da frequência Doppler para a antena 2.

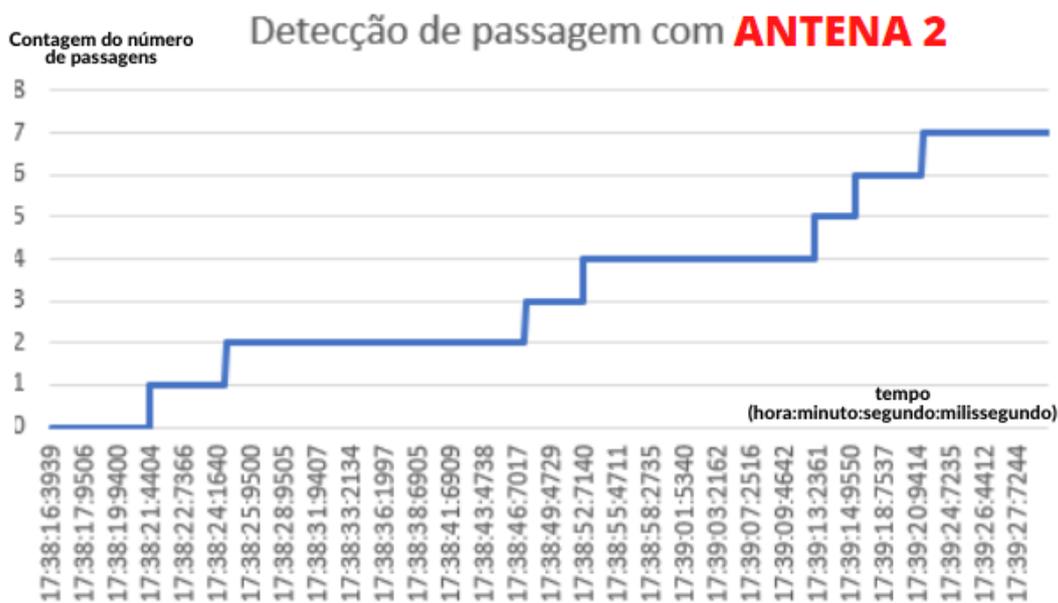


Figura 4.11: Contagem da detecção de passagens pela antena 2.

4.3.1.3 Resultado final obtido do experimento

Agrupando os dados obtidos para as duas antenas, temos os seguintes resultados:

Quantidade de passagens reais: 10.

Quantidade de detecções de passagem para a antena 1: 13.

Quantidade de detecções de passagem para a antena 2: 7.

Erro: 30%

4.3.2 Repetição do experimento

Devido à grande taxa de erro, foi decidido repetir o experimento anterior 10 vezes para se obter dados mais precisos. Para não ocupar muito espaço, o resultado foi resumido na tabela ilustrada na figura 4.1.

Nº do Experimento	Contagem Passagens Antena 1	Contagem Passagens Antena 2
1	13	7
2	12	9
3	7	15
4	10	9
5	13	9
6	8	15
7	12	10
8	8	12
9	16	5
10	10	3

Tabela 4.1: Tabela resumo do resultado do experimento.

Pode-se perceber que em nenhum dos 10 experimentos foi obtido um resultado completamente certo. O experimento mais perto do ideal foi o experimento de número 4, onde foram contabilizados 10 passagens na antena 1 e 9 passagens na antena 2.

4.4 Detecção de pessoas usando informações do RSSI

Como falado anteriormente, a outra abordagem para a detecção da transição de usuários entre os ambientes é utilizando o RSSI, que é um parâmetro que se pode obter diretamente da leitora da Impinj sem a realização de cálculos.

Para a detecção de tags utilizando as leituras do RSSI, foi utilizado o valor de -47 dBm para se considerar a passagem por uma antena. Valores abaixo desse valor poderiam ser considerados um falso positivo e valores acima desses poderiam não ser reconhecidos devido a sensibilidade das antenas, pois a tag teria que passar muito próxima a ela.

A figura 4.12 ilustra o gráfico de um exemplo de leitura do RSSI em relação ao tempo com uma passagem de uma tag por uma antena. Nesta figura pode-se observar claramente a aproximação e o afastamento da tag em relação à antena, gerando uma curva com um pico no instante de ultrapassagem. Quando o valor lido ultrapassa -47 dBm, foi considerado que uma tag está passando em frente à antena.

Da mesma forma que foi feito para o experimento utilizando frequência doppler, por causa da taxa de leitura do equipamento ser de aproximadamente 10 leituras por segundo pode ocorrer uma oscilação do valor lido em torno do valor preestabelecido que indica a passagem da tag. Por conta disto, neste experimento também foi definido um tempo mínimo após a detecção de uma passagem para considerar a detecção de outra passagem.

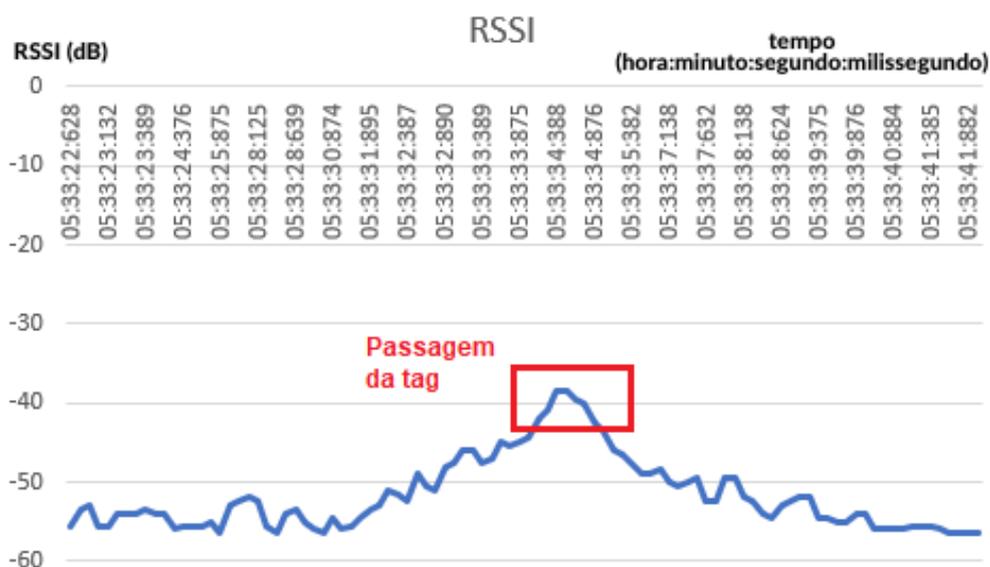


Figura 4.12: Exemplo de leitura do valor RSSI na passagem de uma tag por uma antena.

Após a validação de detecção de uma passagem da tag por meio do RSSI, foram feitos alguns experimentos fazendo 10 passagens com a mesma tag por duas antenas em série.

4.4.1 Experimento de detecção da tag utilizando valor mínimo de -47 dBm de RSSI para considerar passagem

Após a validação de que é possível realizar a detecção de uma passagem da tag utilizando o valor de RSSI, foram feitos alguns experimentos fazendo 10 passagens por duas antena em série com a mesma tag. Os resultados obtidos para a antena 1 e para a antena 2 estão separados nas próximas subseções.

4.4.1.1 Dados do experimento para a antena 1

A figura 4.13 ilustra o gráfico do valor de RSSI em função do tempo para a antena 1. Pode-se observar que no gráfico existem 10 picos que são superiores à -47 dBm, representando um resultado otimista em relação às 10 passagens.

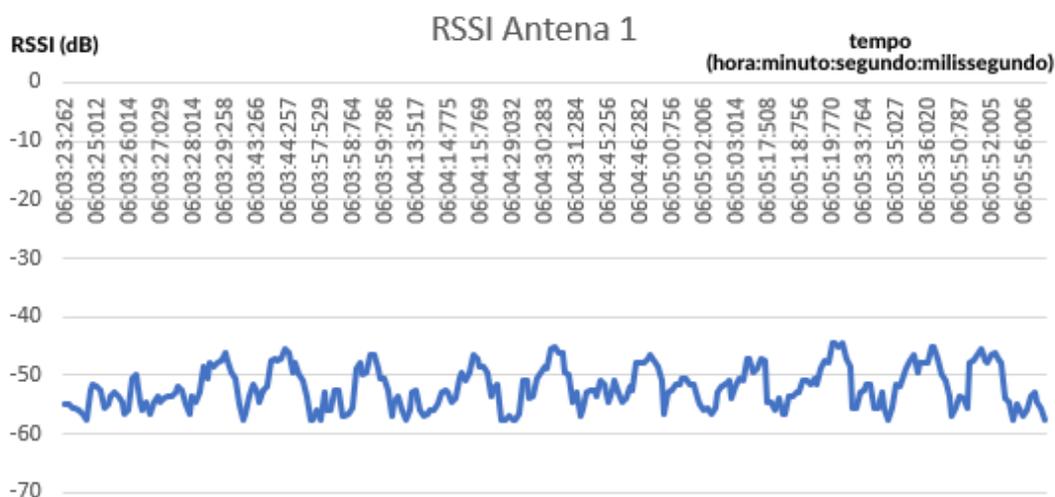


Figura 4.13: Leitura do valor RSSI na passagem da tag pela antena 1 dez vezes.

Analisando o gráfico da detecção de passagens, ilustrado na figura 4.14, se pode observar que as 10 passagens foram contabilizadas corretamente. É possível perceber que houve uma oscilação do sinal, porém não foi tão grande e inconstante igual ao experimento da frequência doppler. Devido a isso, o código realizado foi capaz de realizar a contabilização de forma correta.

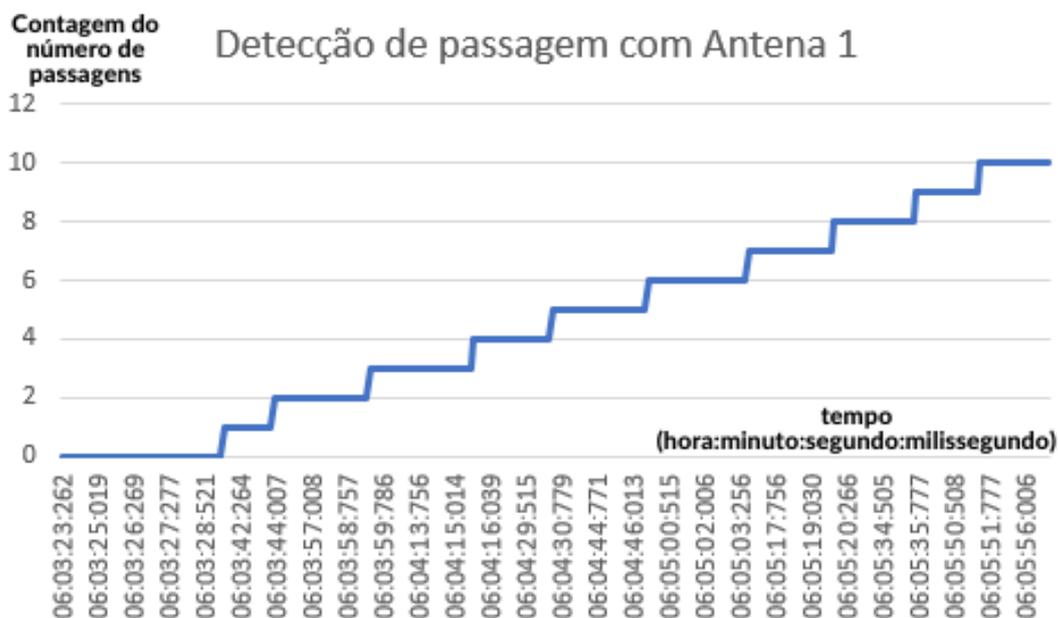


Figura 4.14: Contagem da detecção de passagens pela antena 1.

4.4.1.2 Dados do experimento para a antena 2

A figura 4.15 ilustra o gráfico do RSSI em função do tempo para a antena 2. Neste gráfico também pode-se observar que existem 10 picos que são superiores à -47 dBm.

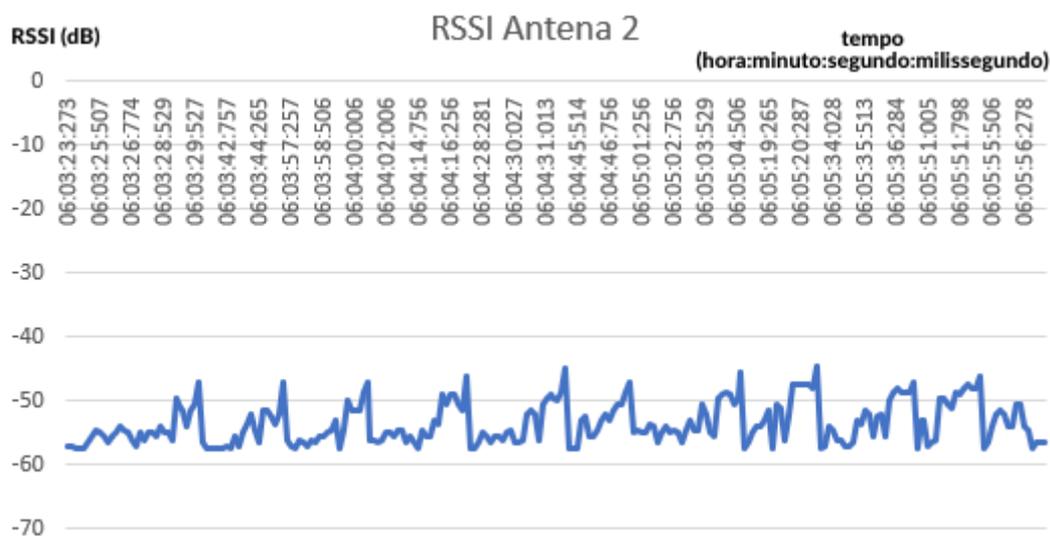


Figura 4.15: Leitura do valor RSSI na passagem da tag pela antena 2 dez vezes.

Analisando o gráfico da detecção de passagens, ilustrado na figura 4.16, se pode observar que as 10 passagens foram contabilizadas corretamente. Da mesma forma que com a antena 1, foi possível perceber que houve oscilação do sinal, porém não o suficiente para causar erros na contagem das passagens da tag pela antena.

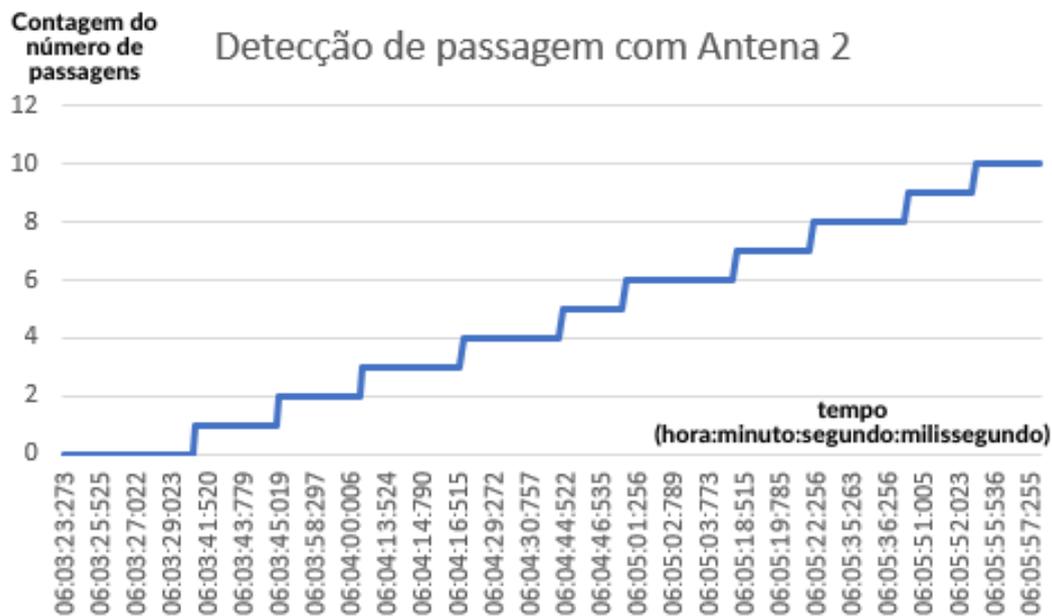


Figura 4.16: Contagem da detecção de passagens pela antena 2.

4.4.1.3 Resultado

Agrupando os dados obtidos para as duas antenas, temos os seguintes resultados:

Quantidade de passagens reais: 10.

Quantidade de detecções de passagem para a antena 1: 10.

Quantidade de detecções de passagem para a antena 2: 10.

Erro: 0%.

4.4.1.4 Repetição do experimento

Para ter um resultado mais preciso, foi repetido o experimento realizando outras 10 passagens com a tag pelas duas antenas. Desta vez, as curvas da antena 1 e antena 2 foram plotadas no mesmo gráfico, assim como a linha com o valor de -47 dBm de RSSI. A figura 4.17 ilustra esse gráfico. As passagens foram realizadas todas no mesmo sentido, de forma que é possível visualizar como ocorrem as transições.

A curva de cor azul representa a leitura da antena 1, a curva de cor vermelha representa a leitura da antena 2 e a linha amarela marca o -47 dBm. Pode-se visualizar que, como a passagem se dá no sentido da antena 1 para a antena 2, o pico de leitura da antena 1 ocorre primeiro que o pico da leitura da antena 2, ambas ultrapassando o valor de -47 dBm representado pela linha amarela. Dessa forma, utilizando esses dados, é possível identificar quando um usuário vai de um ambiente para o outro.

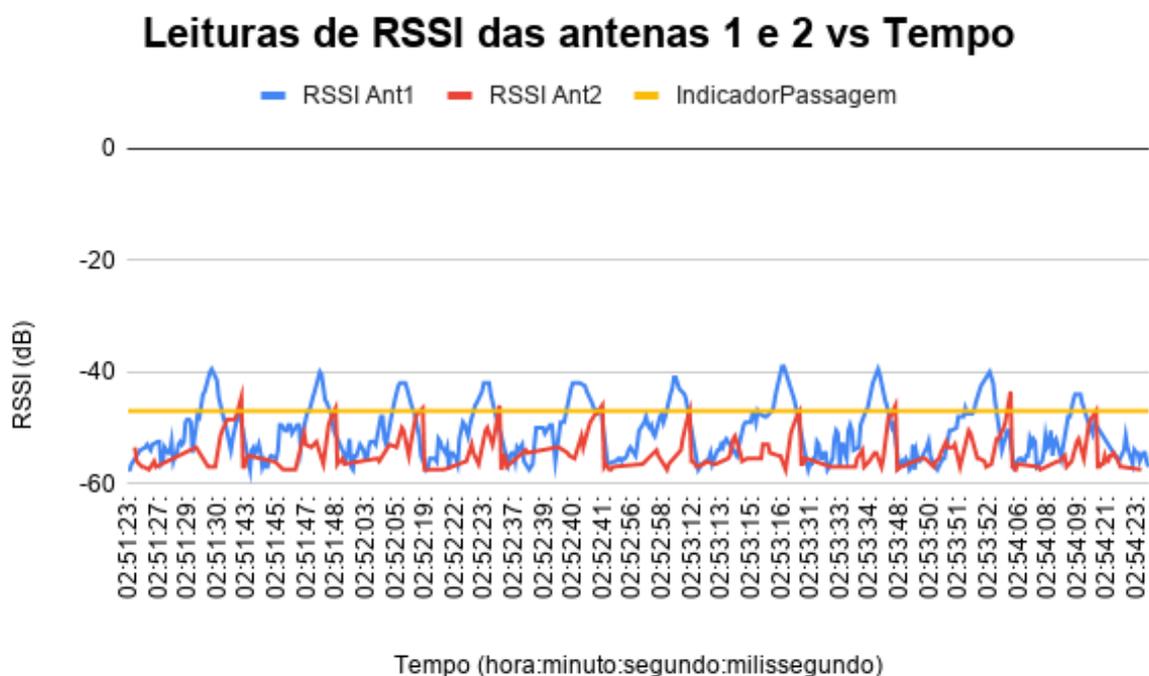


Figura 4.17: Leituras das 10 passagens pela antena 1 e antena 2.

4.5 Testes de transições entre ambientes

```
Leitora endereço: SpeedwayR-10-9F-BB.local
Fernanda realizou uma transicao para o Corredor De Baias
+++++

- Numero de pessoas na Sala principal: 2
  EPC: E200 001B 2609 0146 2580 7745, Nome: Leonardo
  EPC: E200 001B 2609 0146 2770 76FD, Nome: Kenneth
- Numero de pessoas no Corredor de Baias: 1
  EPC: E200 001B 2609 0146 2700 7715, Nome: Fernanda
- Numero de pessoas na Sala de reunioes: 0
- Numero de pessoas no Ambiente Externo: 0

+++++

Leitora endereço: SpeedwayR-10-9F-BB.local
Kenneth realizou uma transicao para o Corredor De Baias
+++++

- Numero de pessoas na Sala principal: 1
  EPC: E200 001B 2609 0146 2580 7745, Nome: Leonardo
- Numero de pessoas no Corredor de Baias: 2
  EPC: E200 001B 2609 0146 2700 7715, Nome: Fernanda
  EPC: E200 001B 2609 0146 2770 76FD, Nome: Kenneth
- Numero de pessoas na Sala de reunioes: 0
- Numero de pessoas no Ambiente Externo: 0

+++++

Leitora endereço: SpeedwayR-10-9F-BB.local
Leonardo realizou uma transicao para o Corredor De Baias
+++++

- Numero de pessoas na Sala principal: 0
- Numero de pessoas no Corredor de Baias: 3
  EPC: E200 001B 2609 0146 2700 7715, Nome: Fernanda
  EPC: E200 001B 2609 0146 2770 76FD, Nome: Kenneth
  EPC: E200 001B 2609 0146 2580 7745, Nome: Leonardo
- Numero de pessoas na Sala de reunioes: 0
- Numero de pessoas no Ambiente Externo: 0

+++++
```

Figura 4.18: Exemplo de 3 transições da Sala Principal para o Corredor de Baias.

Até esta seção, todos os testes foram feitos utilizando apenas uma leitora e um par de antenas. Portanto, foram considerados apenas dois ambientes para os testes, a Sala principal e o Corredor de Baias.

Na figura 4.18, pode-se observar um exemplo da realização de 3 transições de 3 tags diferentes, todas saindo da Sala Principal para o Corredor de Baias. Nesta figura pode-se ver a quantidade de pessoas em cada ambiente, mostrando também o EPC e o nome do usuário associado a cada tag. Inicialmente esses dados são printados no terminal do Visual Studio, porém nas seções

futuras foi implementado uma aplicação de interface gráfica, facilitando a visualização dos dados e proporcionando uma interface mais amigável.

Com a constatação do funcionamento das leitoras e antenas e a validação da estratégia de detecção de passagens, ampliamos o projeto para o monitoramento de 3 ambientes simultaneamente.

Além da ampliação do projeto para múltiplos ambientes, foi implementado um sistema que aciona LEDs. Inicialmente foram utilizados LEDs devido a maior praticidade e facilidade, porém nas seções futuras esses LEDs serão substituídos por relés, que por sua vez podem fazer o acionamento do ar condicionado de cada ambiente.

Os testes foram feitos usando duas leitoras e monitorando três ambientes, que são:

- Área externa
- Sala Principal
- Sala de reuniões

Para os testes de transição em mais de um ambiente, a estratégia adotada foi a utilização dos dados de RSSI. Inicialmente foi considerado que todas as tags começam na área externa. Para realizar a transição das tags entre os ambientes, foi movimentado uma de cada vez.

Cada experimento pode ser separado em dois estágios:

1. Transição das 5 tags da área externa para a sala principal, uma de cada vez (interação 0 a 5)
2. Transição de 3 tags da sala principal para a sala de reuniões, uma de cada vez (interação 6 a 8)

O experimento foi repetido 6 vezes. A seguir estão as imagens dos resultados obtidos, ilustrados nas figuras 4.19, 4.20, 4.21, 4.22, 4.23 e 4.24.

EXP 1						
INTERAÇÃO	TRANSIÇÕES REAIS			TRANSIÇÕES DETECTADAS		
	ÁREA EXTERNA	SALA PRINCIPAL	SALA DE REUNIÕES	ÁREA EXTERNA	SALA PRINCIPAL	SALA DE REUNIÕES
0	5	0	0	5	0	0
1	4	1	0	4	1	0
2	3	2	0	3	2	0
3	2	3	0	2	3	0
4	1	4	0	1	4	0
5	0	5	0	0	5	0
6	0	4	1	0	4	1
7	0	3	2	0	3	2
8	0	2	3	0	2	3

Figura 4.19: Experimento 1 para detecção de passagem de tags entre ambientes.

EXP 2						
INTERAÇÃO	TRANSIÇÕES REAIS			TRANSIÇÕES DETECTADAS		
	ÁREA EXTERNA	SALA PRINCIPAL	SALA DE REUNIÕES	ÁREA EXTERNA	SALA PRINCIPAL	SALA DE REUNIÕES
0	5	0	0	5	0	0
1	4	1	0	4	1	0
2	3	2	0	3	2	0
3	2	3	0	2	3	0
4	1	4	0	1	4	0
5	0	5	0	0	5	0
6	0	4	1	0	4	1
7	0	3	2	0	3	2
8	0	2	3	0	2	3

Figura 4.20: Experimento 2 para detecção de passagem de tags entre ambientes.

EXP 3						
INTERAÇÃO	TRANSIÇÕES REAIS			TRANSIÇÕES DETECTADAS		
	ÁREA EXTERNA	SALA PRINCIPAL	SALA DE REUNIÕES	ÁREA EXTERNA	SALA PRINCIPAL	SALA DE REUNIÕES
0	5	0	0	5	0	0
1	4	1	0	4	1	0
2	3	2	0	3	2	0
3	2	3	0	2	3	0
4	1	4	0	1	4	0
5	0	5	0	0	5	0
6	0	4	1	0	4	1
7	0	3	2	0	3	2
8	0	2	3	0	2	3

Figura 4.21: Experimento 3 para detecção de passagem de tags entre ambientes.

EXP 4						
INTERAÇÃO	TRANSIÇÕES REAIS			TRANSIÇÕES DETECTADAS		
	ÁREA EXTERNA	SALA PRINCIPAL	SALA DE REUNIÕES	ÁREA EXTERNA	SALA PRINCIPAL	SALA DE REUNIÕES
0	5	0	0	5	0	0
1	4	1	0	4	1	0
2	3	2	0	3	2	0
3	2	3	0	2	3	0
4	1	4	0	1	4	0
5	0	5	0	0	5	0
6	0	4	1	0	4	1
7	0	3	2	0	3	2
8	0	2	3	0	2	3

Figura 4.22: Experimento 4 para detecção de passagem de tags entre ambientes.

EXP 5						
INTERAÇÃO	TRANSIÇÕES REAIS			TRANSIÇÕES DETECTADAS		
	ÁREA EXTERNA	SALA PRINCIPAL	SALA DE REUNIÕES	ÁREA EXTERNA	SALA PRINCIPAL	SALA DE REUNIÕES
0	5	0	0	5	0	0
1	4	1	0	4	1	0
2	3	2	0	3	2	0
3	2	3	0	2	3	0
4	1	4	0	1	4	0
5	0	5	0	0	5	0
6	0	4	1	0	5	0
7	0	3	2	0	4	1
8	0	2	3	0	3	2

Figura 4.23: Experimento 5 para detecção de passagem de tags entre ambientes.

EXP 6						
INTERAÇÃO	TRANSIÇÕES REAIS			TRANSIÇÕES DETECTADAS		
	ÁREA EXTERNA	SALA PRINCIPAL	SALA DE REUNIÕES	ÁREA EXTERNA	SALA PRINCIPAL	SALA DE REUNIÕES
0	5	0	0	5	0	0
1	4	1	0	4	1	0
2	3	2	0	3	2	0
3	2	3	0	2	3	0
4	1	4	0	1	4	0
5	0	5	0	0	5	0
6	0	4	1	0	4	1
7	0	3	2	0	3	2
8	0	2	3	0	2	3

Figura 4.24: Experimento 6 para detecção de passagem de tags entre ambientes.

Observando os resultados destes experimentos pode-se ver que apenas no experimento 5 houve erro de detecção de passagem de tag, pois uma passagem não foi identificada.

Comparando estes resultados com os experimentos feitos com Efeito Doppler (ilustrados na seção 4.2), se pode concluir que os dados de RSSI são mais precisos e confiáveis. No total, foram feitas 48 transições entre os ambientes durante este experimento, com apenas um erro de detecção. O erro foi, portanto, de aproximadamente 2%, o que está em uma faixa aceitável para a aplicação deste projeto.

4.6 Acionamento do ar condicionado

O critério utilizado para o controle do ar condicionado foi um indicador obtido através da quantidade de pessoas dentro de um determinado ambiente, de modo que dependendo do gênero da pessoa este indicador tem um valor diferente. Como foi dito anteriormente na seção "Carga térmica" no capítulo 2, geralmente homens emitem mais calor do que mulheres. Por causa disso, caso o usuário associado a tag seja do sexo masculino, ele terá o peso igual a 1 no valor deste indicador para acionamento do relé, caso o usuário seja do sexo feminino, ele terá o peso igual a 0.75 no valor do indicador.

Foi considerado que o indicador para acionamento do relé de um ambiente tenha um valor

superior a 2 para que o ar condicionado seja ligado automaticamente. Por exemplo, suponha que inicialmente existam dois usuários do sexo masculino dentro do mesmo ambiente. Neste caso o ar condicionado permaneceria desligado, pois o indicador teria um valor igual a 2. Caso uma usuária do sexo feminino entre neste mesmo ambiente o ar deveria ligar, pois o indicador é atualizado para o valor de 2.75, e permanecer assim até que algum usuário saia do ambiente e o valor do indicador volte a ser menor ou igual a 2.

Como o LARA não pôde ser usado por causa da pandemia de COVID-19 e por não haver a disposição de um ar condicionado para realizar os testes, foram utilizadas lâmpadas LEDs para simular a ativação de uma carga qualquer. Essas lâmpadas são acionadas por relés, que por sua vez poderiam acionar o ar condicionado.

Para a realização dos testes, foram utilizados os mesmos moldes do experimento anterior (experimento de detecção de passagem de tags). Foram monitorados três ambiente: a área externa, sala principal e sala de reuniões. Nas portas de cada ambiente estão instaladas as leitoras que fazem a detecção de passagem de tags. Cada leitora possui duas antenas que estão posicionadas lado a lado e voltadas para a mesma direção.

As movimentações feitas com as tags foram as mesmas, ou seja, primeiro foi feito a transição de 5 tags da área externa para a sala principal, uma de cada vez. Quando a terceira tag for detectada dentro da sala principal, o ar condicionado deve ser ativado, pois o indicador para acionamento do relé será superior a 2. Em seguida foi feito a transição de 3 tags da sala principal para a sala de reuniões. Na transição da terceira tag, sobrarão apenas duas pessoas na sala principal, portanto o ar condicionado desta sala deve ser desativado. Por outro lado haverão três tags na sala de reuniões, logo o ar condicionado desta sala deve ser ativado.

Foi adicionado uma nova coluna na tabela de resultados do experimento, com o nome "LED ACIONADO". Cada sala possui um LED, que representa o ar condicionado. Ou seja, quando um LED é ativado, significa que o ar condicionado da respectiva sala deve ser acionado. A coluna mostra em qual sala o ar condicionado está ativado.

A seguir estão as tabelas dos testes feitos, ilustrados nas figuras 4.25, 4.26, 4.27, 4.28, 4.29 e 4.30:

INTERAÇÃO	EXP 1						
	TRANSIÇÕES REAIS			ACIONAMENTO LED			
	ÁREA EXTERNA	SALA PRINCIPAL	SALA DE REUNIÕES	ÁREA EXTERNA	SALA PRINCIPAL	SALA DE REUNIÕES	LED ACIONADO
0	5	0	0	5	0	0	NENHUM
1	4	1	0	4	1	0	NENHUM
2	3	2	0	3	2	0	NENHUM
3	2	3	0	2	3	0	SALA PRINCIPAL
4	1	4	0	1	4	0	SALA PRINCIPAL
5	0	5	0	0	5	0	SALA PRINCIPAL
6	0	4	1	0	4	1	SALA PRINCIPAL
7	0	3	2	0	3	2	SALA PRINCIPAL
8	0	2	3	0	2	3	SALA DE REUNIOES

Figura 4.25: Experimento 1 para acionamento do LED em cada ambiente.

INTERAÇÃO	EXP 2						
	TRANSIÇÕES REAIS			ACIONAMENTO LED			
	ÁREA EXTERNA	SALA PRINCIPAL	SALA DE REUNIÕES	ÁREA EXTERNA	SALA PRINCIPAL	SALA DE REUNIÕES	LED ACIONADO
0	5	0	0	5	0	0	NENHUM
1	4	1	0	4	1	0	NENHUM
2	3	2	0	4	1	0	NENHUM
3	2	3	0	3	2	0	NENHUM
4	1	4	0	2	3	0	SALA PRINCIPAL
5	0	5	0	1	4	0	SALA PRINCIPAL
6	0	4	1	1	3	1	SALA PRINCIPAL
7	0	3	2	1	2	2	NENHUM
8	0	2	3	1	1	3	SALA DE REUNIÕES

Figura 4.26: Experimento 2 para acionamento do LED em cada ambiente.

INTERAÇÃO	EXP 3						
	TRANSIÇÕES REAIS			ACIONAMENTO LED			
	ÁREA EXTERNA	SALA PRINCIPAL	SALA DE REUNIÕES	ÁREA EXTERNA	SALA PRINCIPAL	SALA DE REUNIÕES	LED ACIONADO
0	5	0	0	5	0	0	NENHUM
1	4	1	0	4	1	0	NENHUM
2	3	2	0	3	2	0	NENHUM
3	2	3	0	2	3	0	SALA PRINCIPAL
4	1	4	0	1	4	0	SALA PRINCIPAL
5	0	5	0	0	5	0	SALA PRINCIPAL
6	0	4	1	0	4	1	SALA PRINCIPAL
7	0	3	2	0	3	2	SALA PRINCIPAL
8	0	2	3	0	2	3	SALA DE REUNIÕES

Figura 4.27: Experimento 3 para acionamento do LED em cada ambiente.

INTERAÇÃO	EXP 4						
	TRANSIÇÕES REAIS			ACIONAMENTO LED			
	ÁREA EXTERNA	SALA PRINCIPAL	SALA DE REUNIÕES	ÁREA EXTERNA	SALA PRINCIPAL	SALA DE REUNIÕES	LED ACIONADO
0	5	0	0	5	0	0	NENHUM
1	4	1	0	4	1	0	NENHUM
2	3	2	0	3	2	0	NENHUM
3	2	3	0	2	3	0	SALA PRINCIPAL
4	1	4	0	1	4	0	SALA PRINCIPAL
5	0	5	0	0	5	0	SALA PRINCIPAL
6	0	4	1	0	4	1	SALA PRINCIPAL
7	0	3	2	0	3	2	SALA PRINCIPAL
8	0	2	3	0	2	3	SALA DE REUNIÕES

Figura 4.28: Experimento 4 para acionamento do LED em cada ambiente.

INTERAÇÃO	EXP 5						
	TRANSIÇÕES REAIS			ACIONAMENTO LED			
	ÁREA EXTERNA	SALA PRINCIPAL	SALA DE REUNIÕES	ÁREA EXTERNA	SALA PRINCIPAL	SALA DE REUNIÕES	LED ACIONADO
0	5	0	0	5	0	0	NENHUM
1	4	1	0	4	1	0	NENHUM
2	3	2	0	3	2	0	NENHUM
3	2	3	0	3	2	0	NENHUM
4	1	4	0	2	3	0	SALA PRINCIPAL
5	0	5	0	1	4	0	SALA PRINCIPAL
6	0	4	1	1	3	1	SALA PRINCIPAL
7	0	3	2	1	2	2	NENHUM
8	0	2	3	1	1	3	SALA DE REUNIÕES

Figura 4.29: Experimento 5 para acionamento do LED em cada ambiente.

	EXP 6						
	TRANSIÇÕES REAIS			ACIONAMENTO LED			
INTERAÇÃO	ÁREA EXTERNA	SALA PRINCIPAL	SALA DE REUNIÕES	ÁREA EXTERNA	SALA PRINCIPAL	SALA DE REUNIÕES	LED ACIONADO
0	5	0	0	5	0	0	NENHUM
1	4	1	0	4	1	0	NENHUM
2	3	2	0	3	2	0	NENHUM
3	2	3	0	2	3	0	SALA PRINCIPAL
4	1	4	0	1	4	0	SALA PRINCIPAL
5	0	5	0	0	5	0	SALA PRINCIPAL
6	0	4	1	0	4	1	SALA PRINCIPAL
7	0	3	2	0	3	2	SALA PRINCIPAL
8	0	2	3	0	2	3	SALA DE REUNIOES

Figura 4.30: Experimento 6 para acionamento do LED em cada ambiente.

Analisando as tabelas dos resultados do experimento pode-se visualizar que o acionamento dos LEDs ocorreu de acordo com o que foi previsto, ou seja, acionou sempre que o indicador para acionamento do relé foi superior a 2, ou seja, houve pelo menos três tags em um mesmo ambiente, e desativou quando houve menos tags. Logo, não ocorreram erros para o acionamento dos LEDs. A parte crítica é, portanto, a detecção de passagens de tags entre os ambientes.

4.7 Interface gráfica

Utilizando o Windows Forms, foi feito um aplicativo com uma interface gráfica para a visualização das transições entre os ambientes, contagem de pessoas em cada ambiente, verificação do acionamento dos relés, dados dos usuários em cada ambiente e outras informações de interesse.

A interface foi criada para facilitar a visualização dos testes e proporcionar uma maior praticidade, uma vez que a visualização das transições por meio do terminal de comandos dificulta o entendimento e não é agradável ao usuário.

Na figura 4.31 pode-se observar a tela inicial da interface, de modo que é possível ver as listas com EPC e nome associado a tag para cada ambiente. Também é possível visualizar o valor do indicador utilizado para o acionamento do LED e também se o relé está ou não acionado. No canto esquerdo inferior é possível visualizar a quantidade total de pessoas.

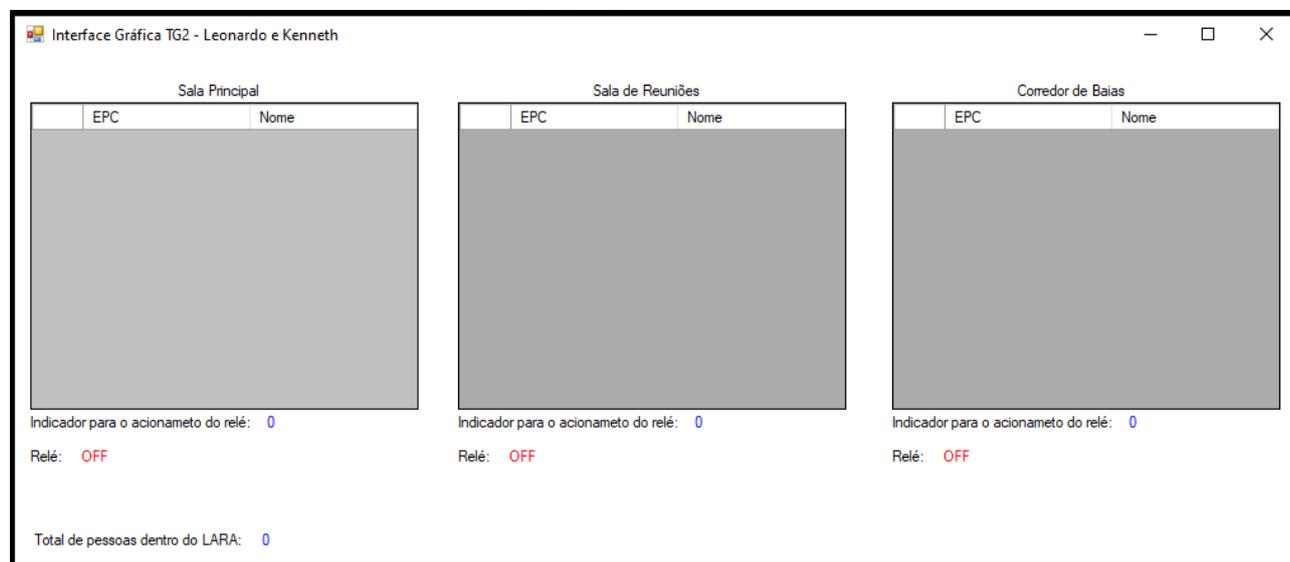


Figura 4.31: Tela inicial da interface sem pessoas dentro dos ambientes.

Para o teste das funcionalidades desse aplicativo foram feitas transições de três tags entre dois ambientes. Inicialmente foi feita a transição de uma pessoa do ambiente externo para a sala principal, como mostra a figura 4.32.

Dessa forma, pode-se observar que na caixa referente à sala principal uma tag é representada, mostrando seu EPC e nome da pessoa. Por ser um homem, o indicador utilizado para o acionamento do relé é atualizado para 1, assim como o total de pessoas, que está representado no canto inferior esquerdo da interface.

Em seguida foi feita a transição de outra pessoa do ambiente externo para a sala principal, como pode ser visto na figura 4.33. Por se tratar de uma mulher, foi somado o valor de 0.75 no indicador utilizado para o acionamento do relé. O total de pessoas é atualizado para 2.

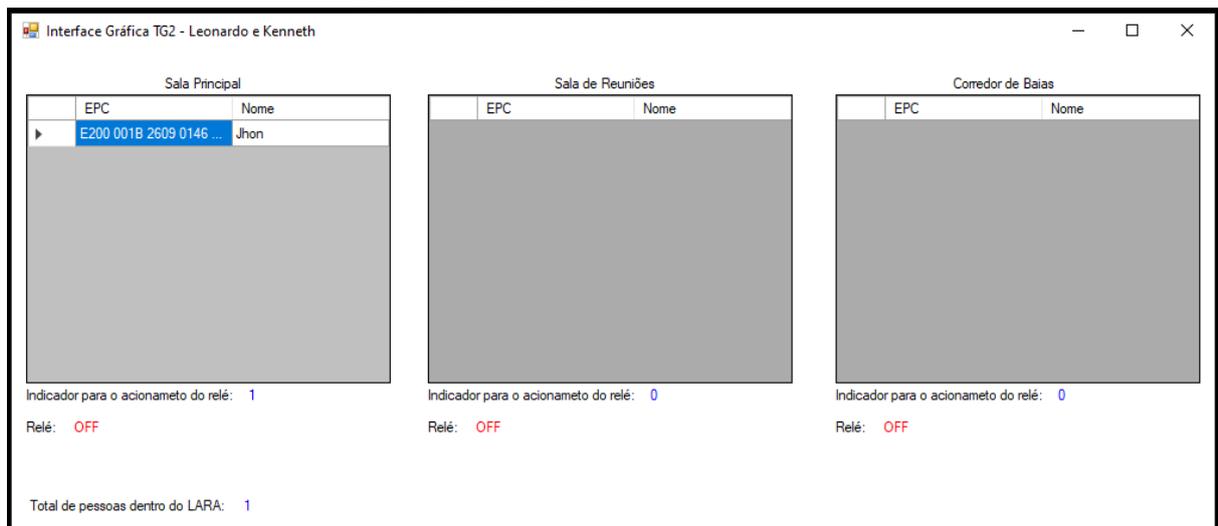


Figura 4.32: Interface após transição de uma pessoa do ambiente externo para a sala principal.

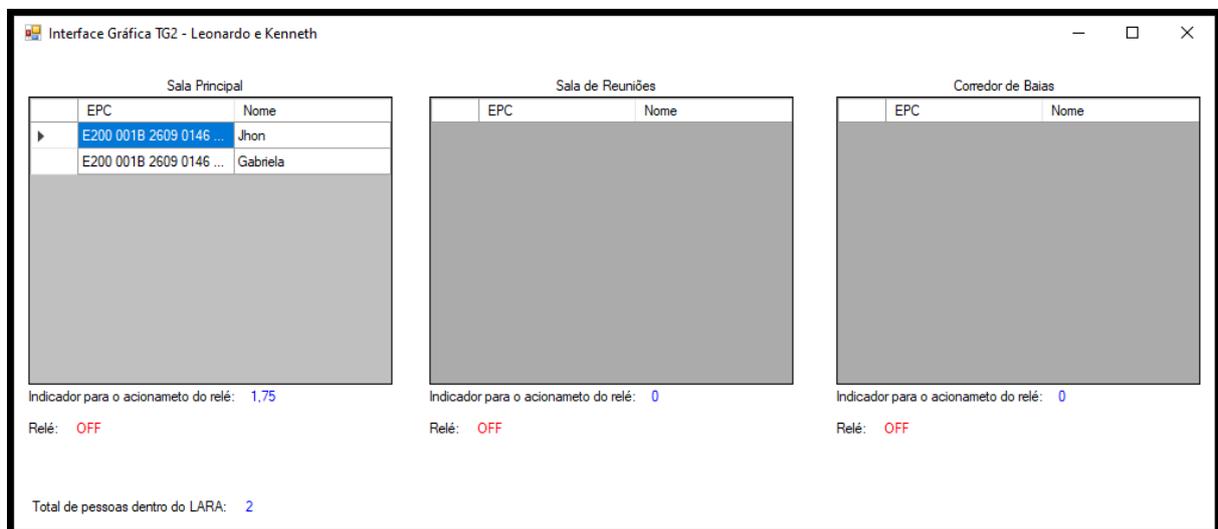


Figura 4.33: Interface com duas pessoas na sala principal.

Então, foi feita a transição de mais uma pessoa do ambiente externo para a sala principal, como pode ser visto na figura 4.34. Por ser uma mulher, é somado 0,75 no indicador utilizado para o acionamento do relé, chegando a um total de 2,5. Como o indicador é superior a 2, o relé é acionado e texto é alterado para "ON", indicando que o relé na sala principal foi acionado. O total de pessoas é atualizado para 3.

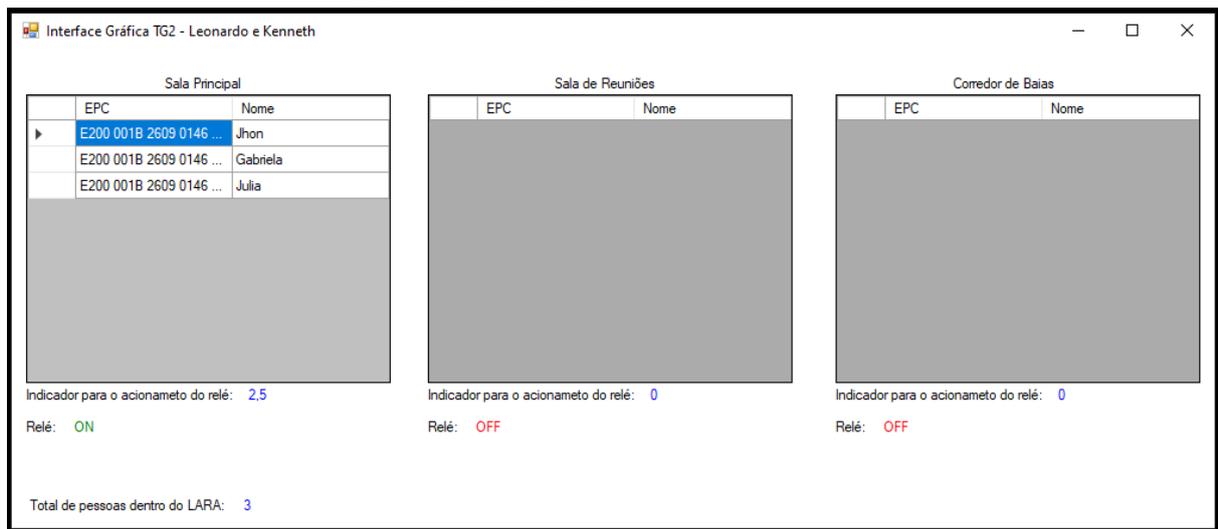


Figura 4.34: Interface com três pessoas na sala principal.

Após isso, foi feita a transição de uma pessoa da sala principal para o corredor de baias, como pode ser visto na figura 4.35. Dessa forma, os indicadores utilizados para o acionamento dos relés são atualizados em ambos os ambientes. Como o indicador na sala principal cai para 1.5, o relé é desligado e passa para "OFF". Como não houve mais transições do ambiente externo para algum dos ambientes internos, o número total de pessoas permanece em 3.

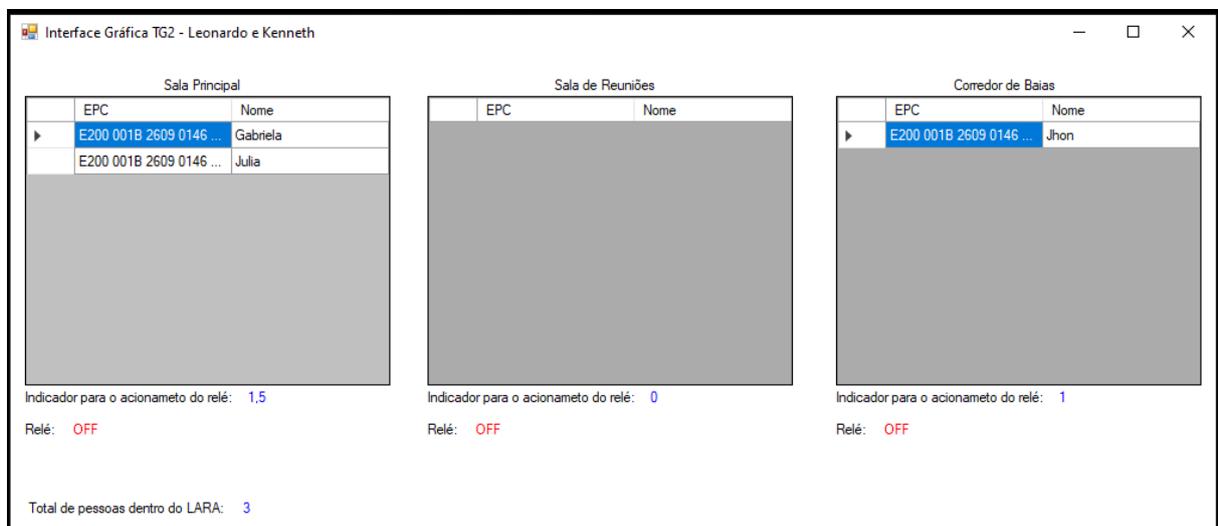


Figura 4.35: Interface com a transição de uma pessoa da sala principal para o corredor de baias.

Em seguida, foi feita a transição de mais uma pessoa da sala principal para o corredor de baias, como pode ser visto na figura 4.36. Novamente, os indicadores são atualizados e como ambos não são maiores que 2, não ocorre o acionamento dos relés. O total de pessoas também permanece o mesmo pois não houve novas pessoas nos ambientes, como era esperado.

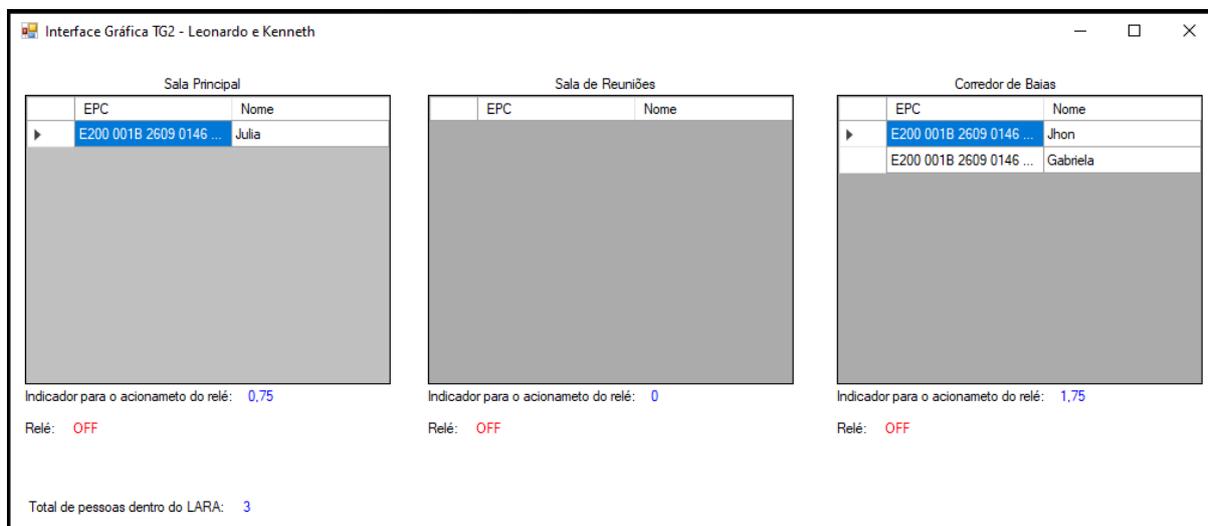


Figura 4.36: Interface com a transição de duas pessoas da sala principal para o corredor de baias.

Por fim, foi realizado a transição da última pessoa na sala principal para o corredor de baias, como é visto na figura 4.37. Dessa forma, os dados são atualizados e como o indicador utilizado para o acionamento do relé do corredor de baias é superior a 2, ocorre o seu acionamento e o texto muda para "ON".

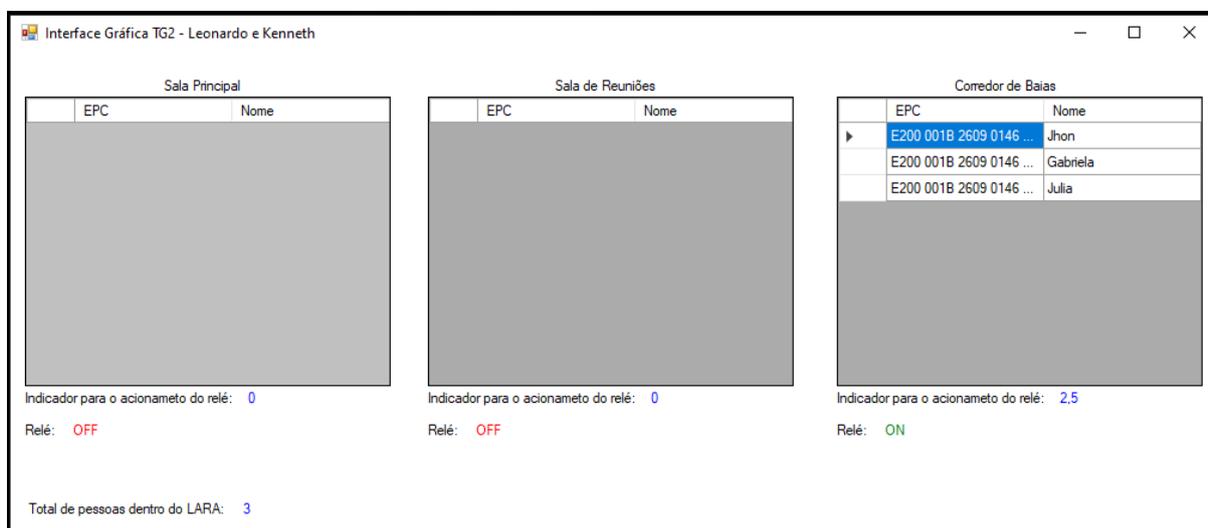


Figura 4.37: Interface com a transição de três pessoas da sala principal para o corredor de baias.

Portanto, pode-se perceber que o aplicativo está funcionando corretamente e da forma esperada, mostrando em tempo real as transições e dados importantes, como o total de pessoas e o acionamento do relé, de modo a facilitar a visualização dos testes realizados neste trabalho.

4.8 Teste de movimentação de tag com parada no caminho

Foi realizado um teste para observar o comportamento do sistema caso algum usuário passasse pela primeira antena de uma leitora mas desistisse da passagem na metade do caminho, dando meia volta. O objetivo deste teste é verificar se o programa interpretaria ou não como uma transição. Se trata de um teste relevante pois pode acontecer de um usuário decidir ir somente até a divisória dos ambientes mas sem a intenção de realizar uma transição.

Foi constatado que uma das antenas detecta a tag, no entanto isso não acontece com a outra. Isso está ilustrado na figura 4.38. O gráfico se refere à leitura feita pela antena 2.

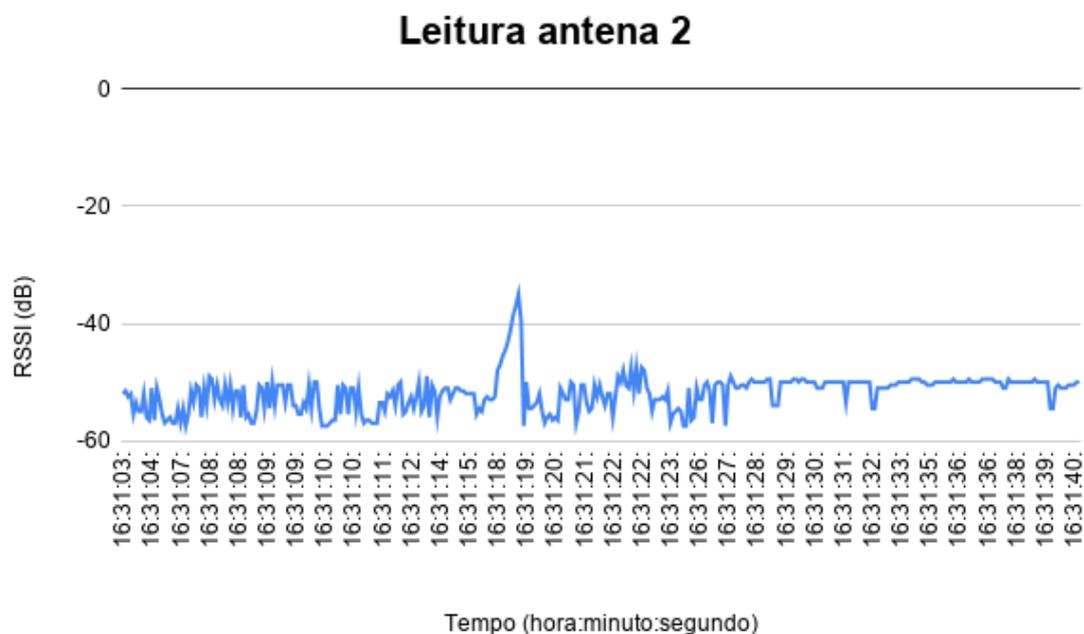


Figura 4.38: Experimento de movimentação de tag com parada no caminho.

Pela figura 4.38 observa-se que entre os segundos 18 e 19 o valor do RSSI ultrapassa o valor de -47 dB e portanto há uma detecção de passagem da tag pela antena 2. A antena 1, no entanto, não detectou a tag e por isso não foi ilustrada.

Analisando os dados das figuras 4.39 e 4.40 constatou-se que não foi detectada nenhuma transição entre a área externa e a sala principal. Ambos os gráficos permaneceram inalterados, indicando que não houve mudança no número de pessoas dentro de cada ambiente.

Portanto, o resultado obtido deste experimento foi o resultado esperado. Como o usuário se moveu até a metade do caminho, somente uma das antenas detectou a passagem e portanto não houve a transição da tag para o outro ambiente, como era esperado.

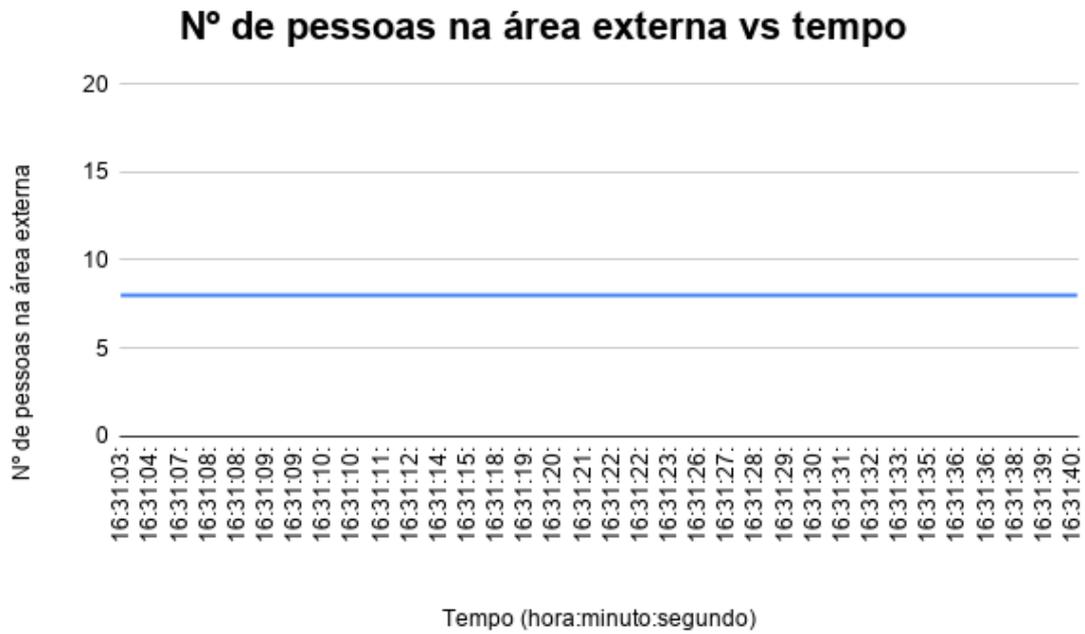


Figura 4.39: Gráfico do número de pessoas dentro do ambiente externo durante o experimento.

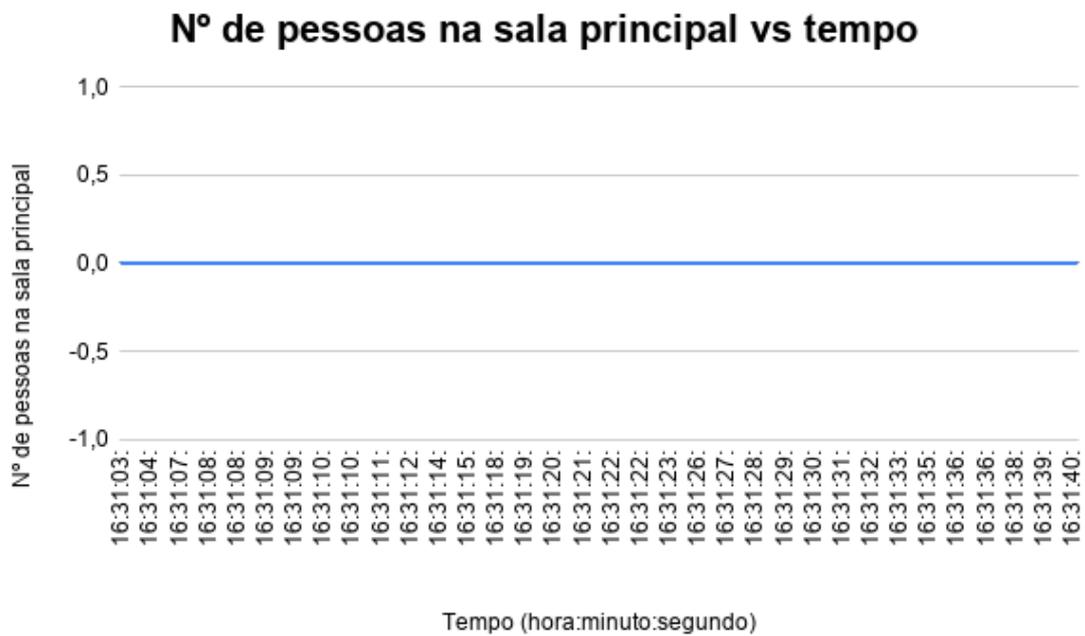


Figura 4.40: Gráfico do número de pessoas dentro da sala principal durante o experimento.

4.9 Passagem de 2 tags simultaneamente

Foi feito um teste para verificar a detecção de tags no caso de passagem de duas tags simultaneamente pela mesma leitora. O objetivo deste experimento é verificar se o programa lidaria de forma correta com duas pessoas tentando passar ao mesmo tempo de um ambiente para outro.

Inicialmente há 8 tags na área externa. A transição foi feita usando 2 tags para transitar simultaneamente da área externa para a sala principal e os dados obtidos pela leitora (nomeada "leitora 2") estão ilustrados nas figuras 4.41, 4.42, 4.43 e 4.44.



Figura 4.41: Leitura RSSI para passagem de 2 tags simultaneamente.

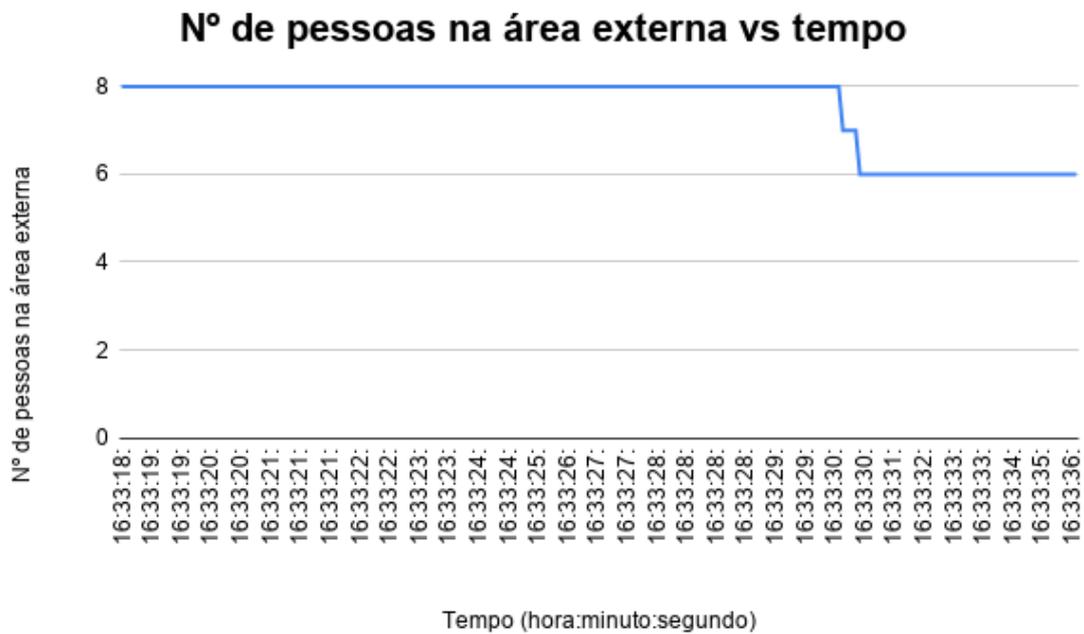


Figura 4.42: Contagem de número de pessoas na área externa.

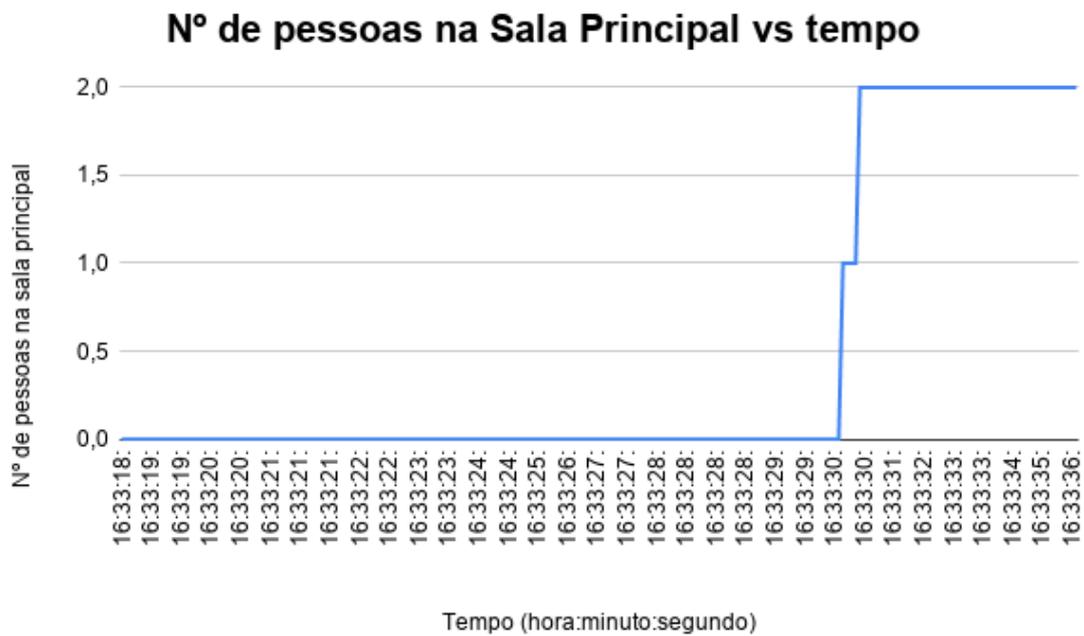


Figura 4.43: Contagem de número de pessoas na sala principal.

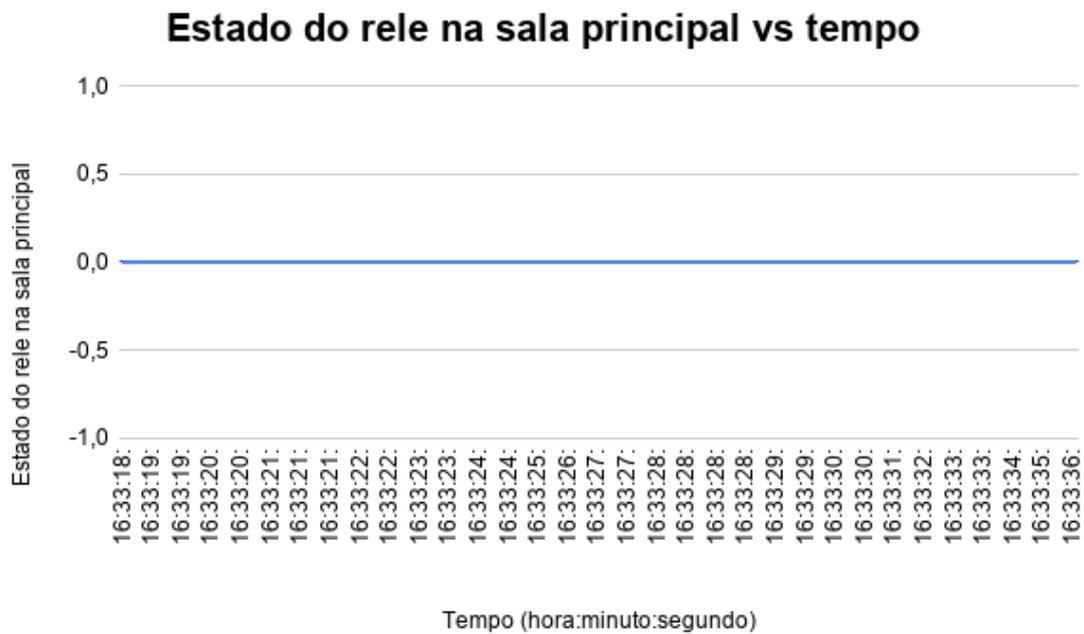


Figura 4.44: Acionamento do rele na sala principal.

Analisando os dados, observa-se que a passagem ocorreu entre os segundos 30 e 31, quando a segunda antena da leitora detectou a passagem das tags. Isso ocorre porque a detecção de passagem é considerada apenas quando a tag passa pelas duas antenas à uma proximidade mínima, provocando um valor RSSI mínimo preestabelecido.

Pode-se ver que o relé da sala principal (representando o ar condicionado) não foi ativado, pois apenas duas pessoas estão na sala e portanto o valor do indicador para acionamento do relé não satisfaz o peso mínimo para o acionamento.

4.10 Passagem de 3 tags simultaneamente

O experimento anterior foi repetido usando 3 tags ao invés de 2. Portanto, inicialmente haviam 8 tags posicionadas na área externa e foram usadas 3 tags para transitarem para a sala principal simultaneamente. No fim, deve haver 5 tags na área externa e 3 tags na sala principal.

Este experimento está ilustrado pelas figuras 4.45, 4.46, 4.47 e 4.48.



Figura 4.45: Leitura RSSI para passagem de 3 tags simultaneamente.



Figura 4.46: Contagem de número de pessoas na área externa.

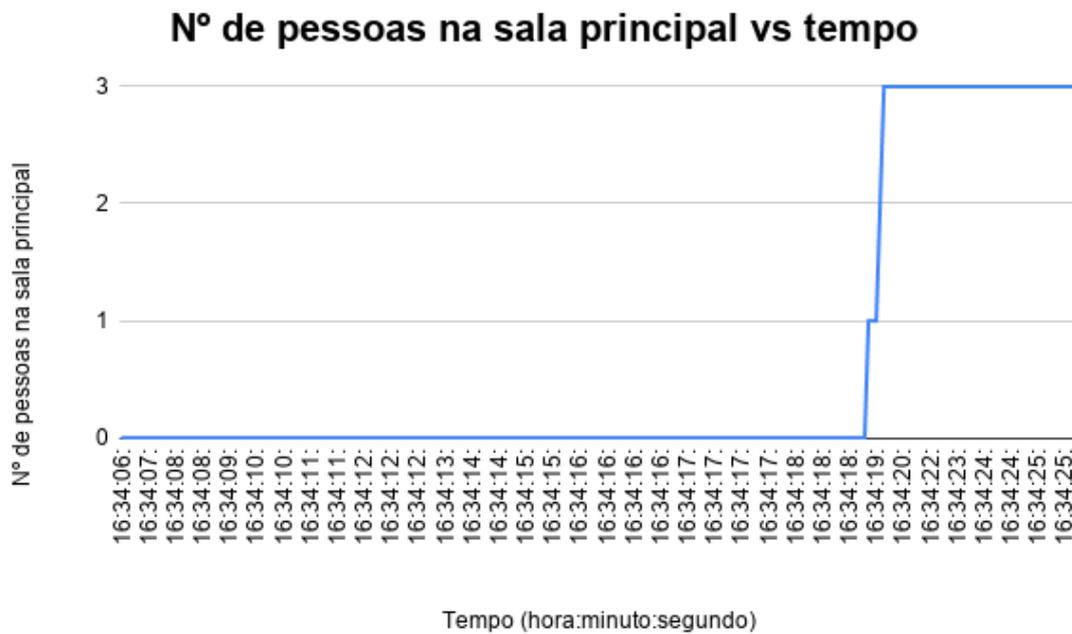


Figura 4.47: Contagem de número de pessoas na sala principal.

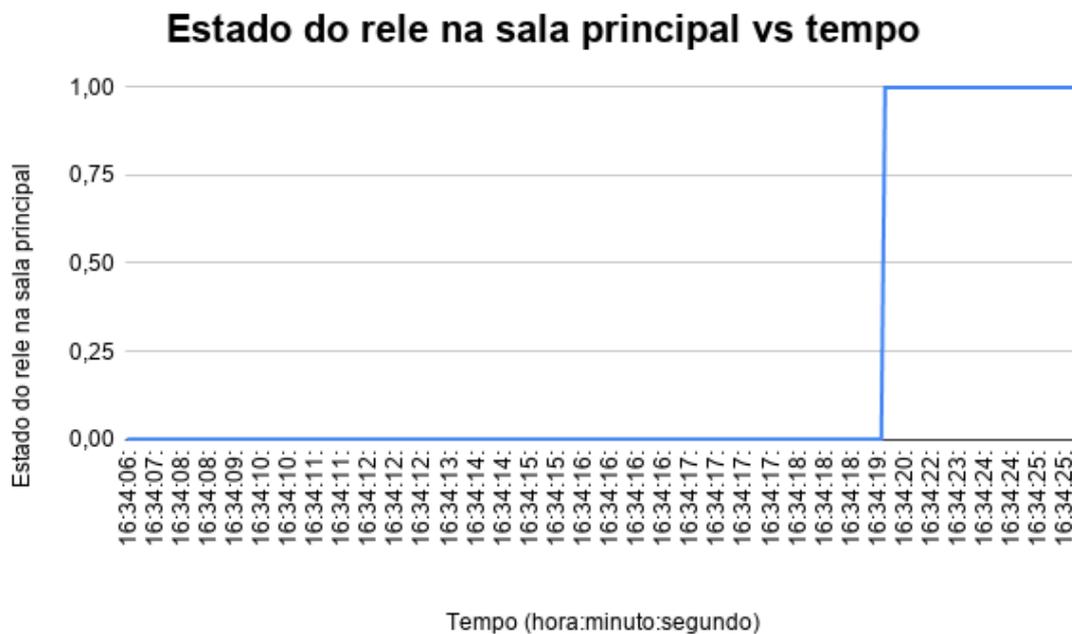


Figura 4.48: Acionamento do relé na sala principal.

Analisando os dados, observa-se que a passagem ocorreu entre os segundos 17 e 20, quando a segunda antena da leitora detectou a passagem das tags. Isso ocorre porque a detecção de passagem é considerada apenas quando a tag passa pelas duas antenas à uma proximidade mínima.

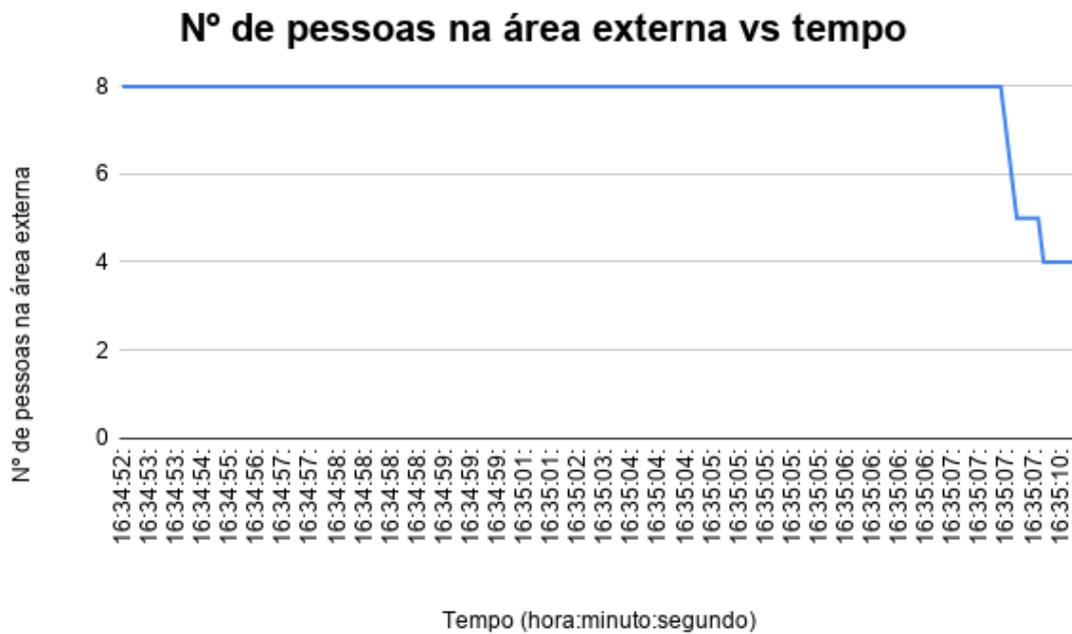


Figura 4.50: Contagem de número de pessoas na área externa.

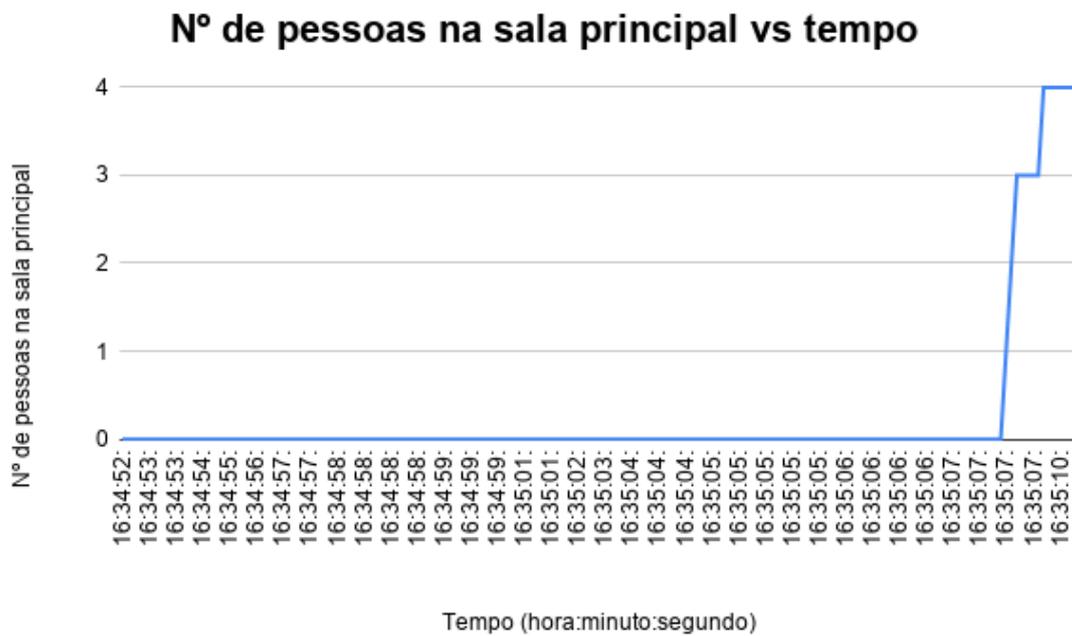


Figura 4.51: Contagem de número de pessoas na sala principal.

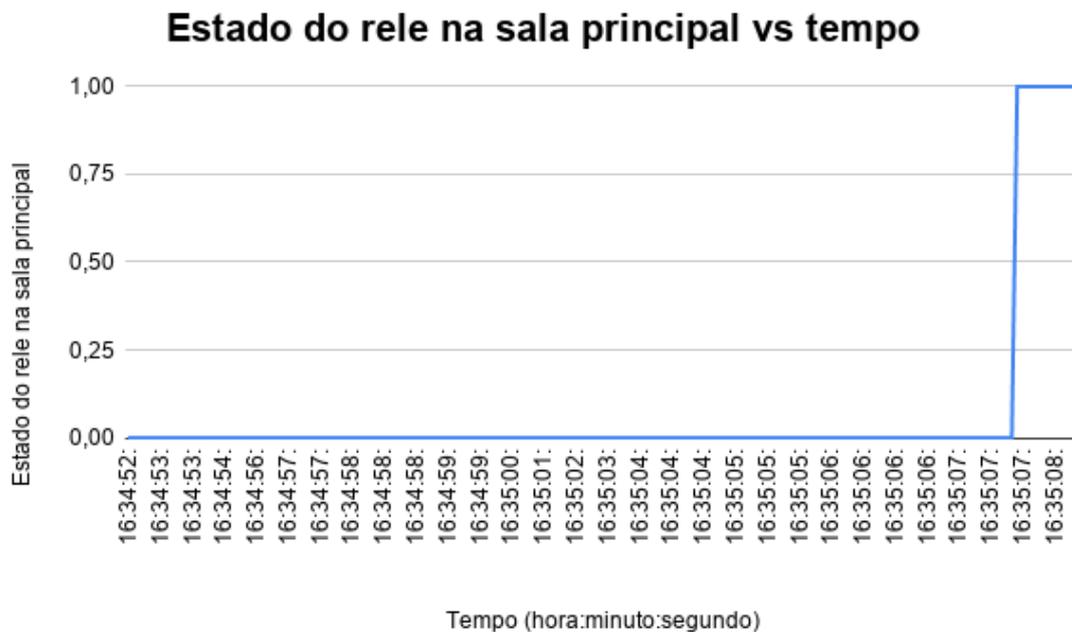


Figura 4.52: Acionamento do rele na sala principal.

Analisando os dados, observa-se que a passagem ocorreu no entre os segundos 5 e 7, quando a segunda antena da leitora termina de detectar as passagens das tags. Isso ocorre porque a detecção de passagem é considerada apenas quando a tag passa pelas duas antenas à uma proximidade mínima.

Pode-se ver que o relé da sala principal (representando o ar condicionado) foi ativado, pois quatro pessoas estão na sala e portanto o indicador para acionamento do relé satisfaz o peso mínimo para o acionamento, sendo ativado no segundo 07.

4.12 Passagem de 5 tags simultaneamente

O experimento anterior foi repetido usando 5 tags ao invés de 4. Portanto, inicialmente haviam 8 tags posicionadas na área externa e foram usadas 5 tags para transitarem para a sala principal simultaneamente. No fim, deve haver 3 tags na área externa e 5 tags na sala principal.

Este experimento está ilustrado pelas figuras 4.53, 4.54, 4.55 e 4.56.

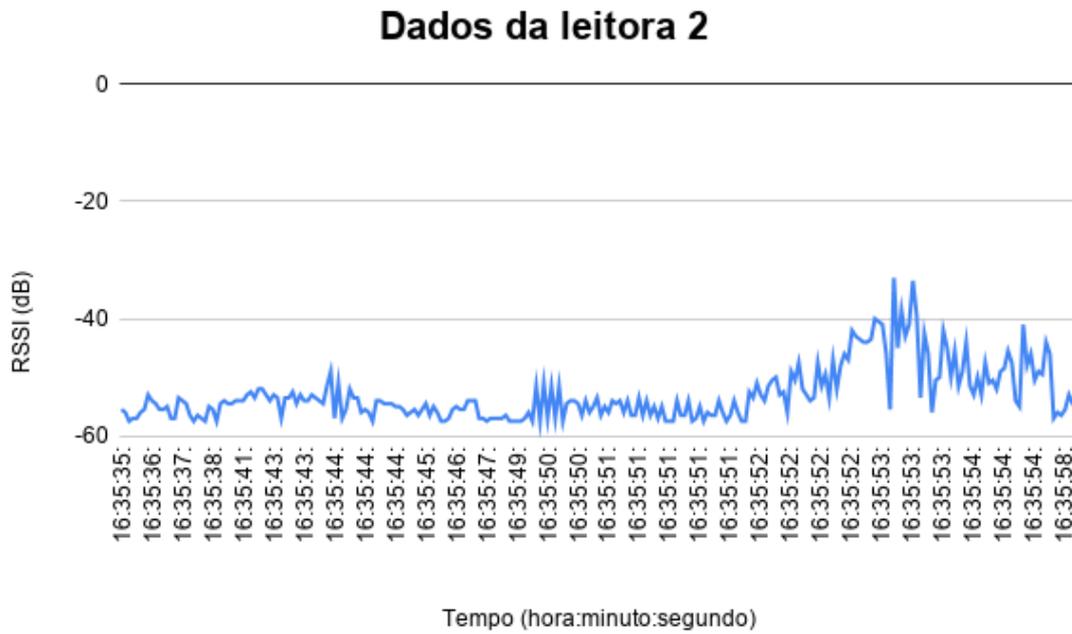


Figura 4.53: Leitura RSSI para passagem de 5 tags simultaneamente.

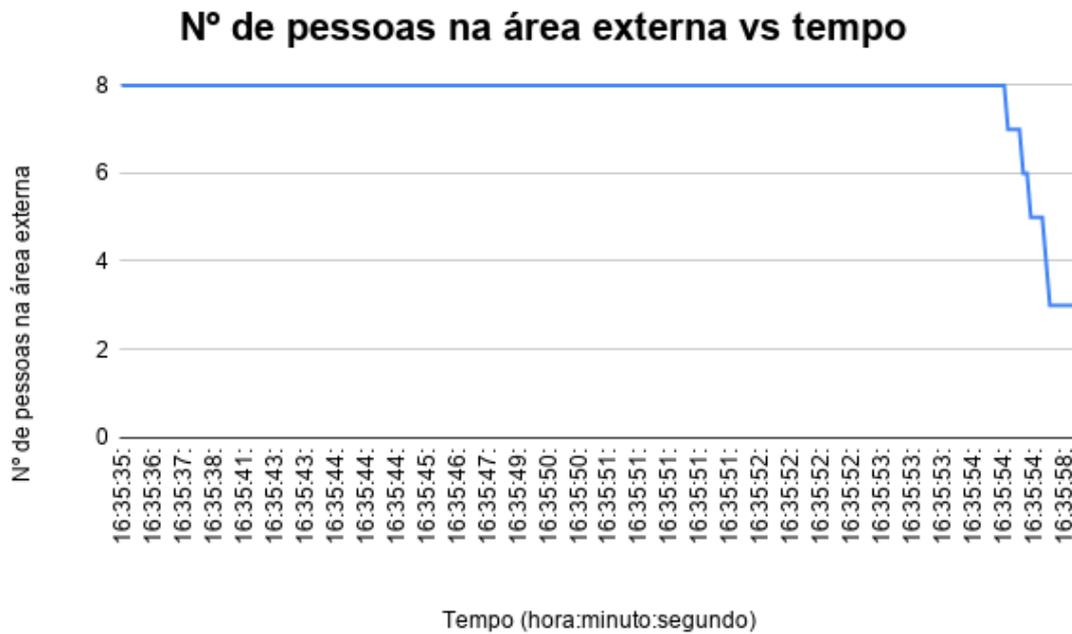


Figura 4.54: Contagem de número de pessoas na área externa.

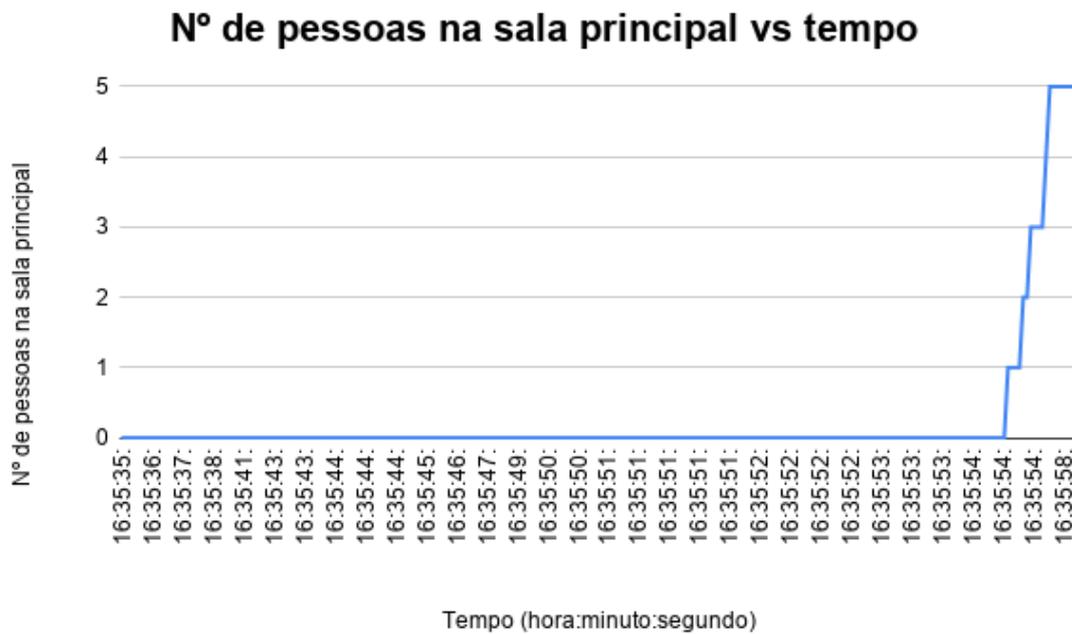


Figura 4.55: Contagem de número de pessoas na sala principal.

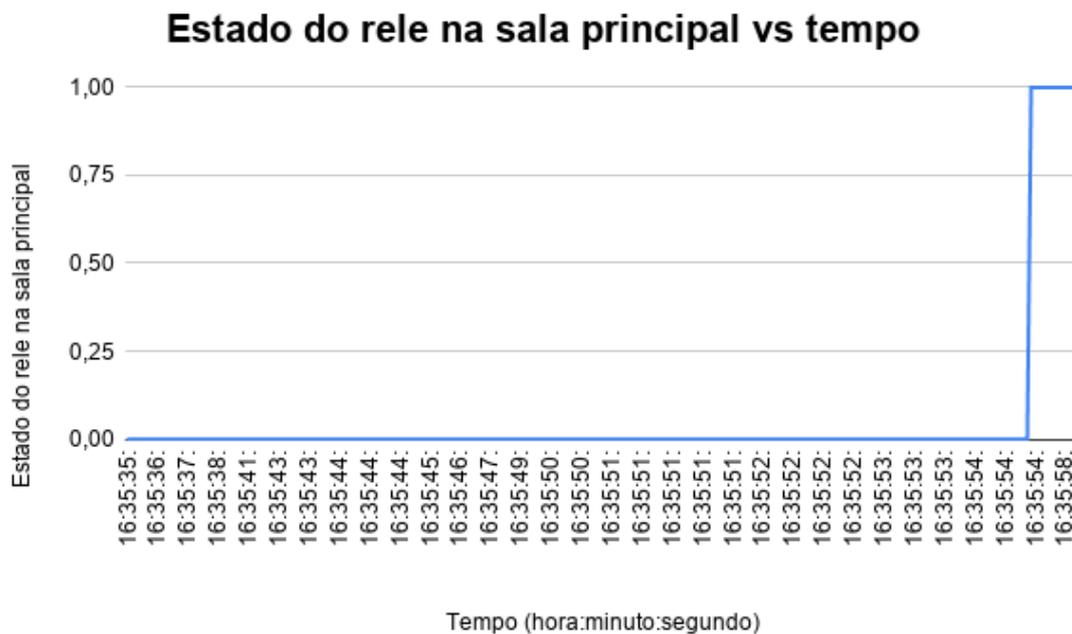


Figura 4.56: Acionamento do rele na sala principal.

Analisando os dados, observa-se que a passagem ocorreu no entre os segundos 53 e 54, quando a segunda antena da leitora termina de detectar as passagens das tags. Isso ocorre porque a detecção de passagem é considerada apenas quando a tag passa pelas duas antenas à uma proximidade

mínima.

Pode-se ver que o rele da sala principal (representando o ar condicionado) foi ativado, pois cinco pessoas estão na sala e portanto o indicador para acionamento do relé satisfaz o peso mínimo para o acionamento, sendo ativado no segundo 54.

Percebe-se que à medida que aumentamos o número de tags que passam simultaneamente fica mais difícil diferenciar os picos de leitura entre as leitoras. No entanto a contagem e detecção de passagens continua precisa.

4.13 Duas passagens simultâneas em dois ambientes diferentes

Foi feito um teste realizando duas passagens simultâneas em duas salas diferentes para observar como seria o resultado na leitura e detecção de passagens. O objetivo era verificar o funcionamento do sistema para passagens simultâneas em ambientes diferentes e com mais de uma tag em cada passagem.

Para realizar o experimento, inicialmente todas as tags foram posicionadas na área externa. Então quatro tags foram deslocadas para a sala principal. A partir desta posição, duas tags foram deslocadas para a sala de reuniões e outras duas tags foram deslocadas para o corredor de baias, sendo todos estes deslocamentos feitos ao mesmo tempo.

O resultado está ilustrado nas figuras 4.57, que representa a quantidade de pessoas que se encontram na sala principal ao decorrer do tempo 4.58, que representa a quantidade de pessoas que se encontram na sala de reuniões ao decorrer do tempo, e 4.59, que representa a quantidade de pessoas que se encontram no corredor de baias ao decorrer do tempo.



Figura 4.57: Número de pessoas na sala principal.

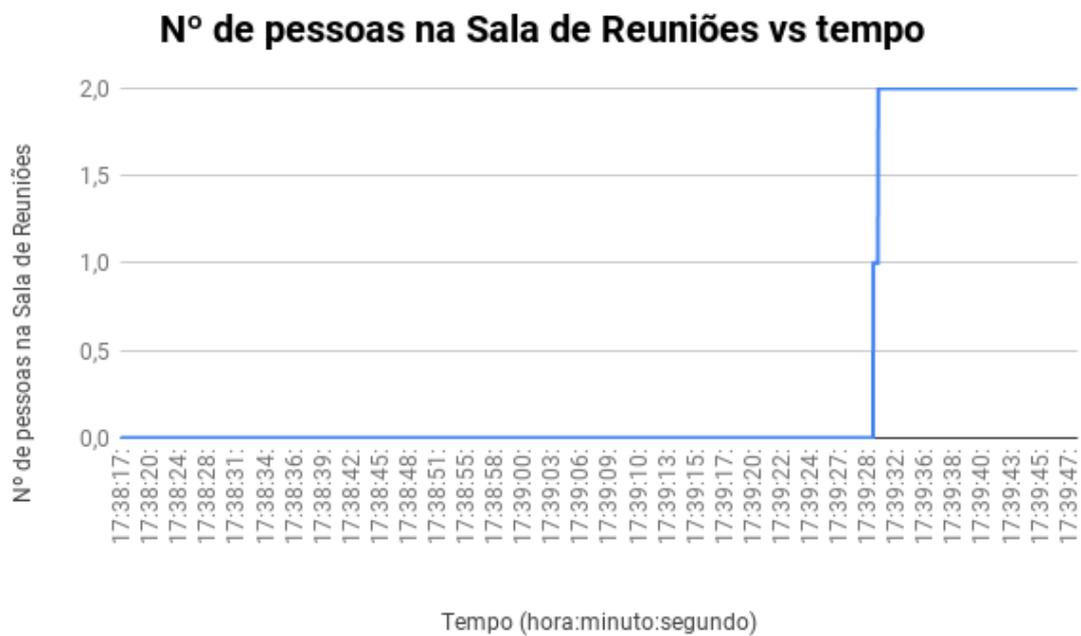


Figura 4.58: Número de pessoas na sala de reuniões.

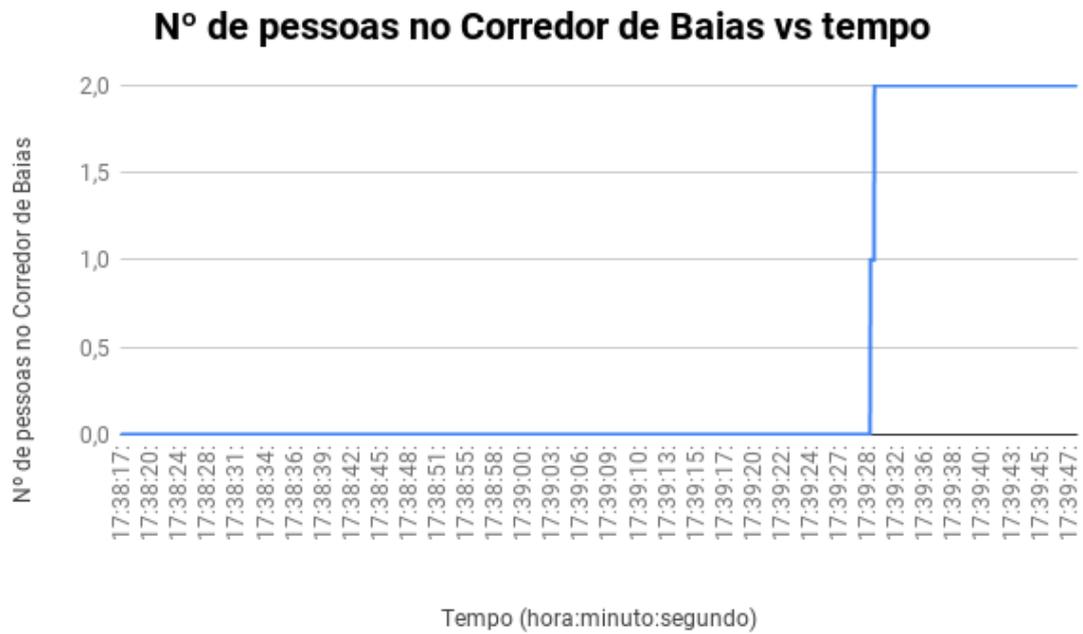


Figura 4.59: Nº de pessoas no corredor de baias.

Pode-se observar por este experimento que as detecções e contagens foram feitas corretamente. Isso mostra que o sistema funciona para múltiplas passagens simultâneas, podendo ser aplicada na monitoração de múltiplas salas simultaneamente.

4.14 Testes de leitura de acordo com distância

O objetivo dos testes a seguir é verificar como a precisão na detecção e contagens de passagens de tags é afetada pela distância da tag em relação à antena. Para isso, foi realizado dois tipos de teste: um fazendo passagens com a tag orientada para a antena e outra com passagens com a tag orientada perpendicularmente à antena.

4.14.1 Testes com tag orientada com a face para a antena

A disposição da tag e antena durante a passagem neste experimento está ilustrado na figura 4.60. A tag está com a face voltada para a antena, e esta está voltada para a tag.

Foram feitos testes com diferentes distâncias entre a tag e as antenas. Para cada experimento, foram realizadas 10 transições no total, ou seja, 5 transições saindo do ambiente e 5 transições entrando no ambiente. Inicialmente se iniciou o experimento com a tag dentro da Sala Principal. Nos gráficos abaixo (ilustrados nas figuras 4.61, 4.62, 4.63, 4.64, 4.65 e 4.66) é possível visualizar o número de tags dentro de cada ambiente, evidenciando as transições realizadas.

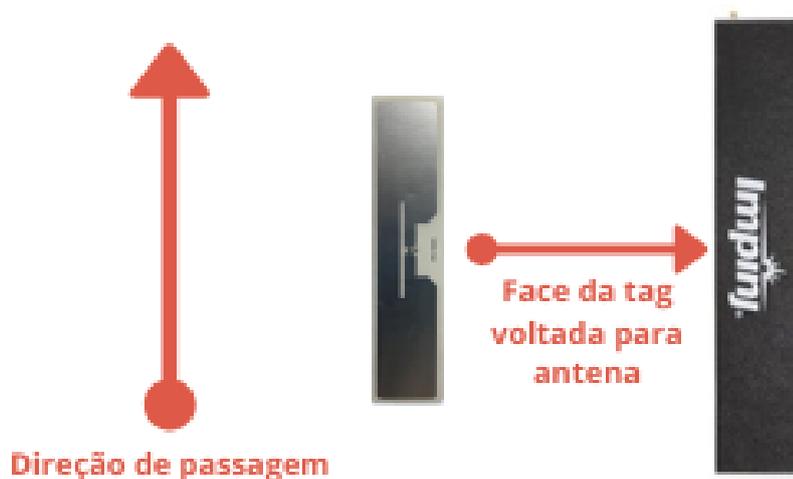


Figura 4.60: Ilustração da orientação da tag na passagem com a face voltada para a antena.

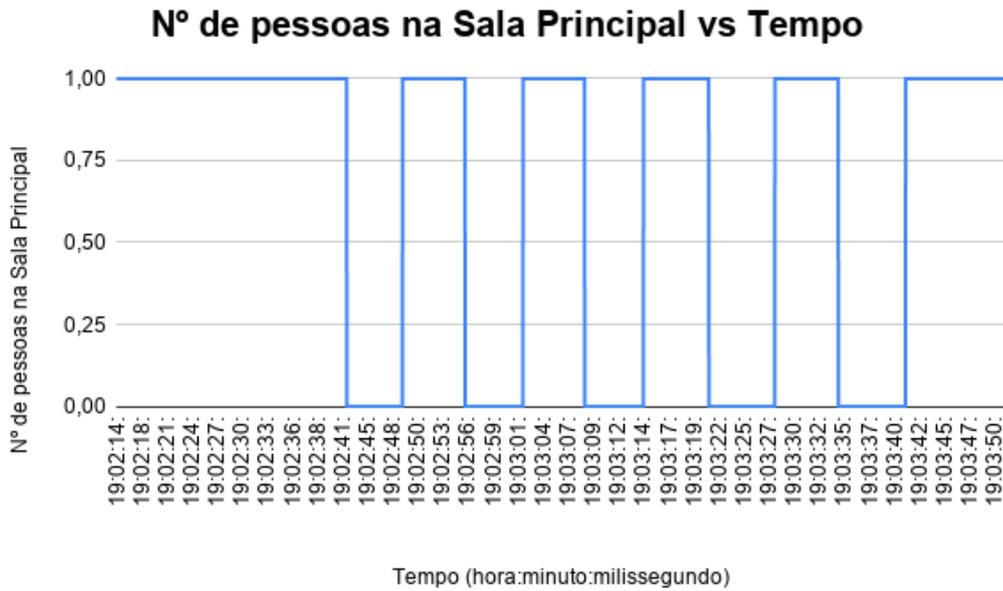


Figura 4.61: Passagem a 30 cm de distância da antena com a tag orientada para a antena.

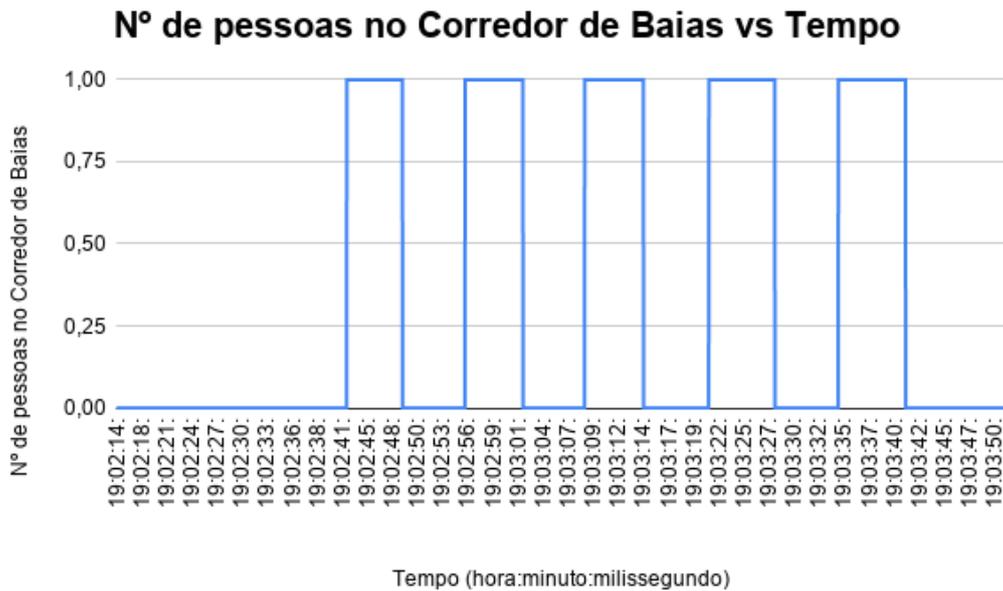


Figura 4.62: Passagem a 30 cm de distância da antena com a tag orientada para a antena.

Resultados para experimento com distância de 30 cm:

Número de transições reais: 10

Número de transições detectadas: 10

Precisão: 100%

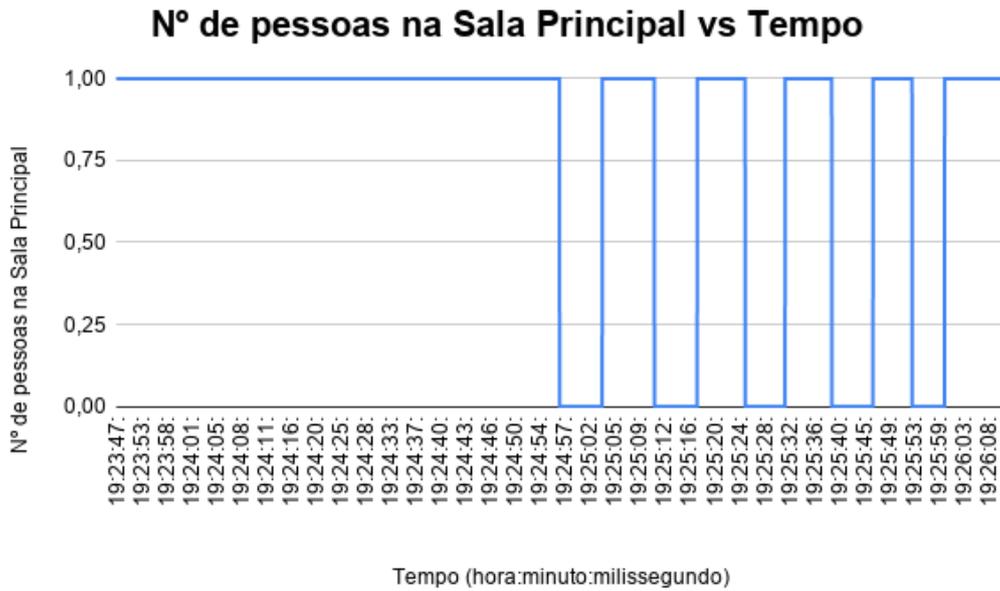


Figura 4.63: Passagem a 60 cm de distância da antena com a tag orientada para a antena.

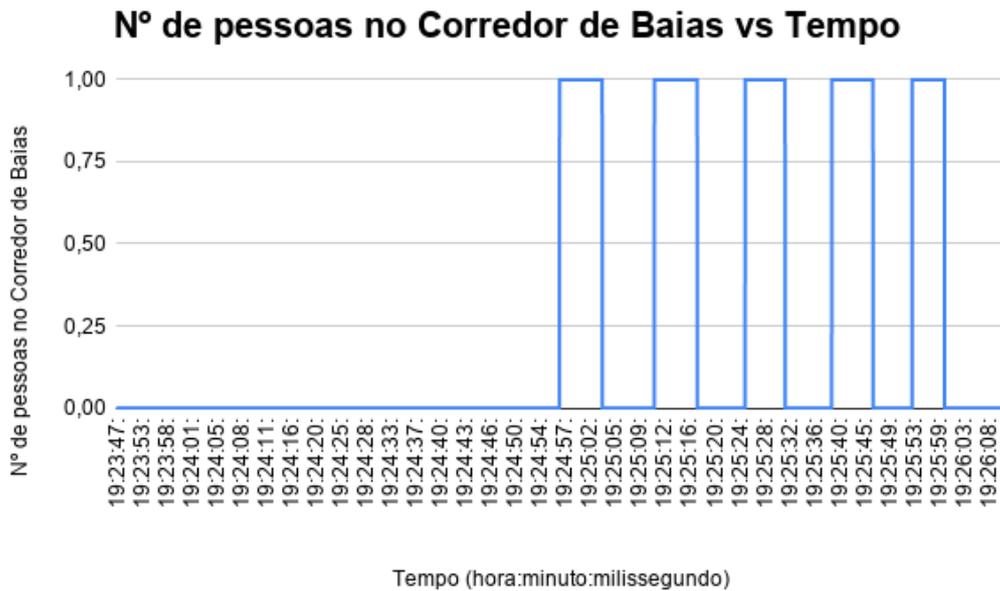


Figura 4.64: Passagem a 60 cm de distância da antena com a tag orientada para a antena.

Resultados para experimento com distância de 60 cm:

Número de transições reais: 10

Número de transições detectadas: 10

Precisão: 100%

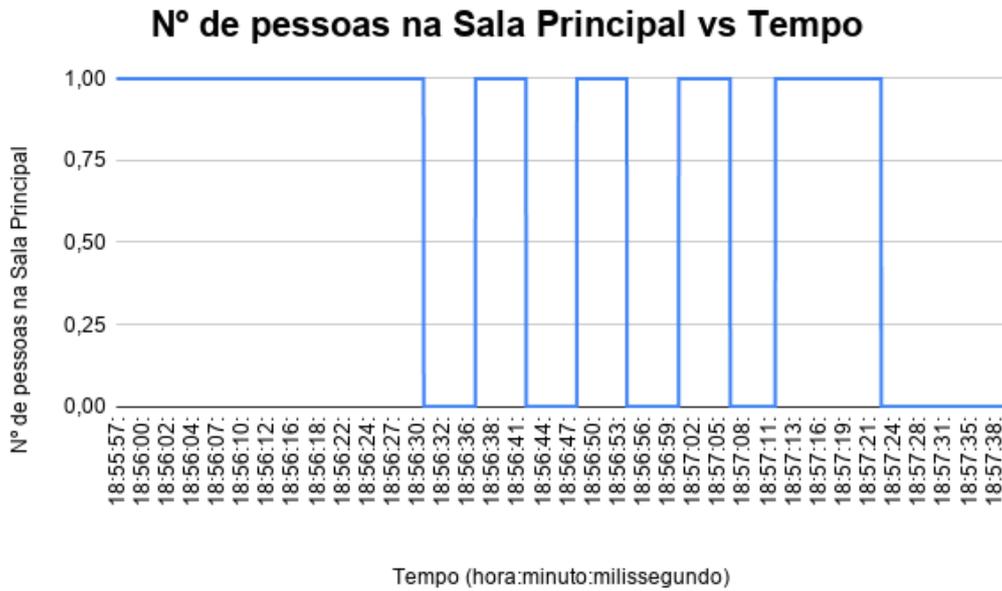


Figura 4.65: Passagem a 90 cm de distância da antena com a tag orientada para a antena.

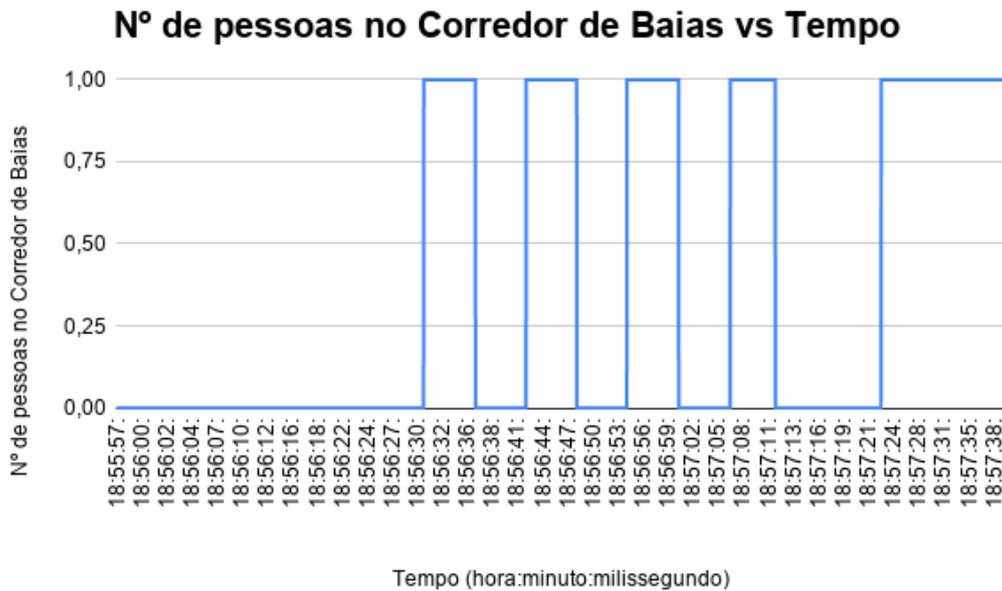


Figura 4.66: Passagem a 90 cm de distância da antena com a tag orientada para a antena.

Resultados para experimento com distância de 90 cm:

Número de transições reais: 10

Número de transições detectadas: 9

Precisão: 90%

Portanto, conclui-se que para distâncias de até 90 centímetros a detecção de transição é precisa se a tag estiver com a face orientada para a antena. Considerando que a abertura de uma porta geralmente não passa de 90 centímetros, assumiu-se que se trata de um resultado bastante satisfatório.

4.14.2 Testes com tag orientada perpendicularmente à antena

A disposição da tag e antena durante as transições neste experimento está ilustrado na figura 4.67. A tag está com a face voltada à direção perpendicular à antena, e esta está voltada para a tag. Foi repetido o experimento anterior, alterando somente a orientação da face da tag. Os resultados dos experimentos estão ilustrados nas figuras 4.68, 4.69, 4.70, 4.71, 4.72 e 4.73.



Figura 4.67: Ilustração da orientação da tag na passagem.

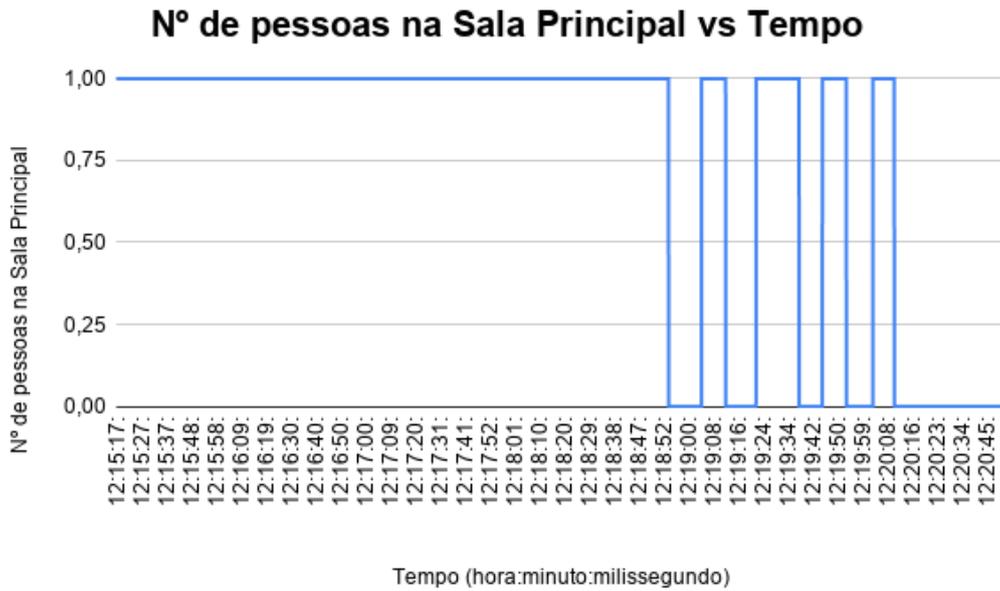


Figura 4.68: Passagem a 30 cm de distância da antena com a tag orientada perpendicularmente à antena.

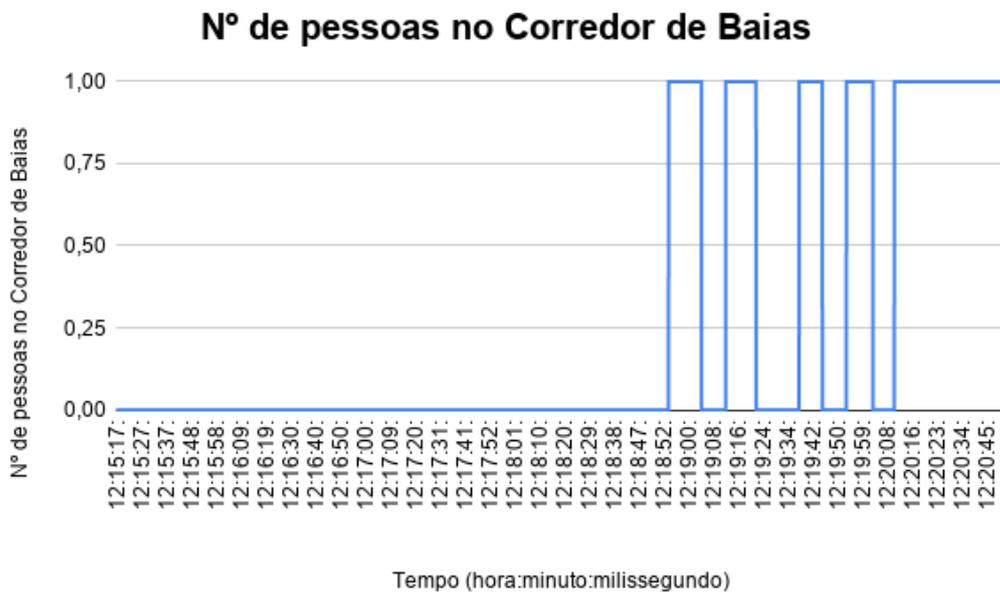


Figura 4.69: Passagem a 30 cm de distância da antena com a tag orientada perpendicularmente à antena.

Resultados para experimento com distância de 30 cm:

Número de transições reais: 10

Número de transições detectadas: 9

Precisão: 90%



Figura 4.70: Passagem a 60 cm de distância da antena com a tag orientada perpendicularmente à antena.

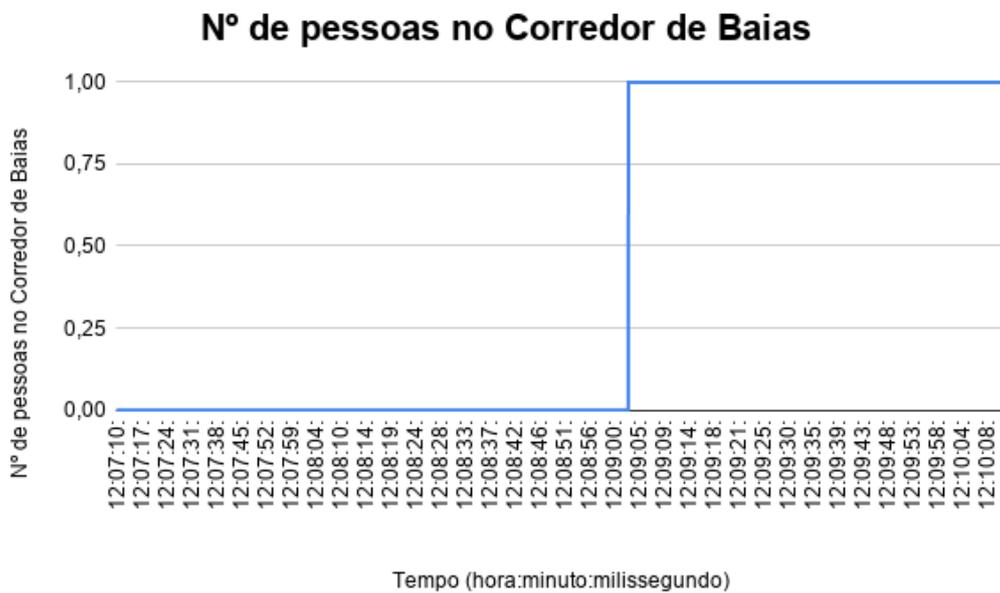


Figura 4.71: Passagem a 60 cm de distância da antena com a tag orientada perpendicularmente à antena.

Resultados para experimento com distância de 60 cm:

Número de transições reais: 10

Número de transições detectadas: 1

Precisão: 10%

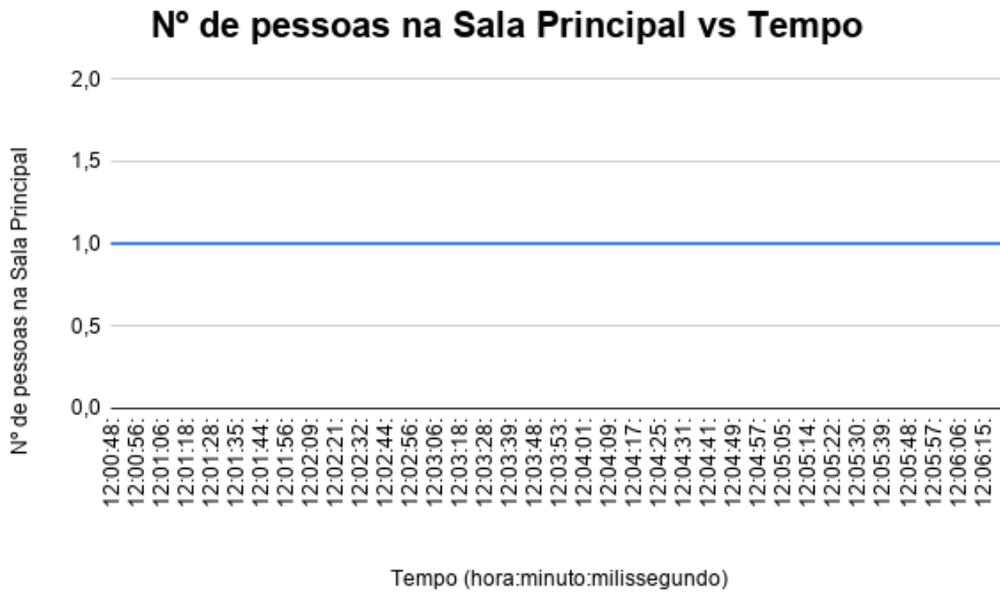


Figura 4.72: Passagem a 90 cm de distância da antena com a tag orientada perpendicularmente à antena.

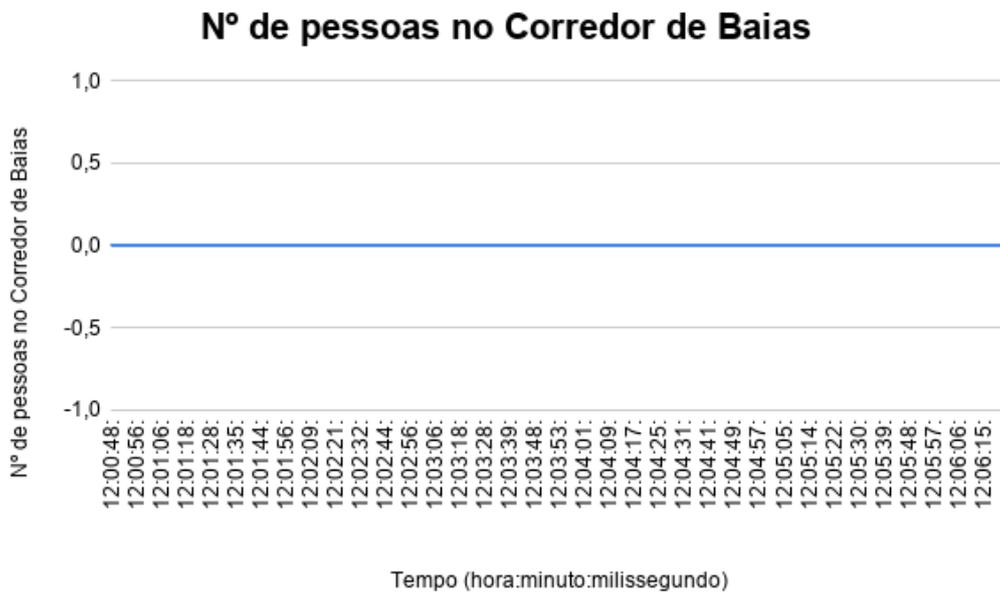


Figura 4.73: Passagem a 90 cm de distância da antena com a tag orientada perpendicularmente à antena.

Resultados para experimento com distância de 90 cm:

Número de transições reais: 10

Número de transições detectadas: 0

Precisão: 0%

Como observado pelas figuras acima, o teste feito com passagem da tag à 60 centímetros da antena detectou apenas uma transição de dez feitas, enquanto o teste feito com passagem da tag à 90 centímetros da antena não detectou nenhuma transição. Houve uma redução drástica entre o teste com a distância de 30 centímetros para o teste com distância de 60 centímetros. Devido a isto, decidiu-se realizar mais um teste com passagem da tag à 45 centímetros da antena para ver o resultado, uma distância intermediária entre 60 cm e 30 cm. O resultado obtido está ilustrado nas figuras 4.74 e 4.75:

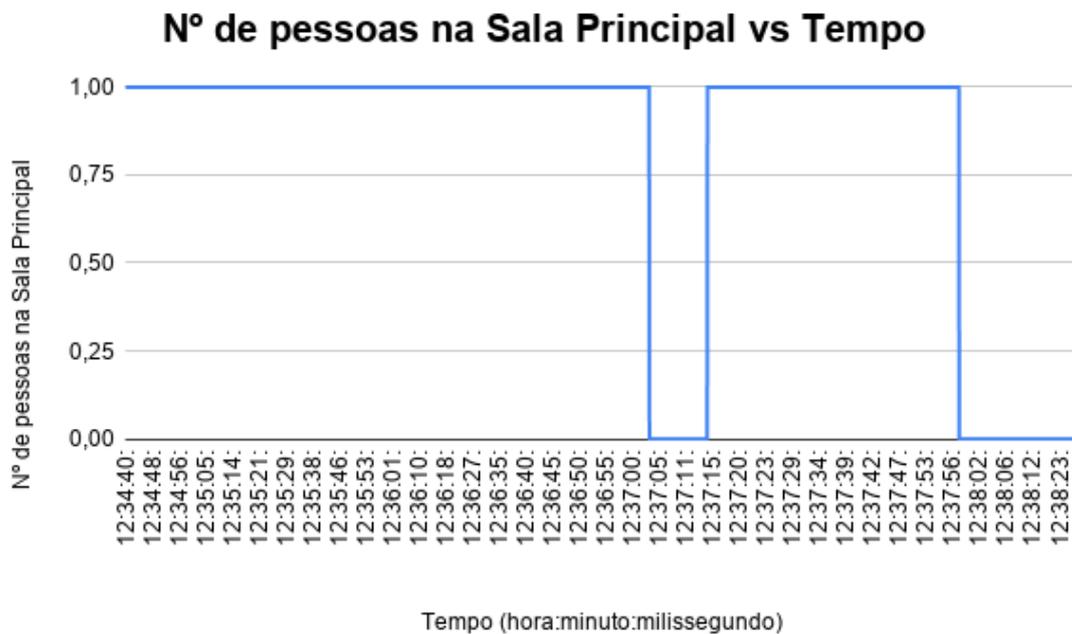


Figura 4.74: Passagem a 45 cm de distância da antena com a tag orientada perpendicularmente à antena.

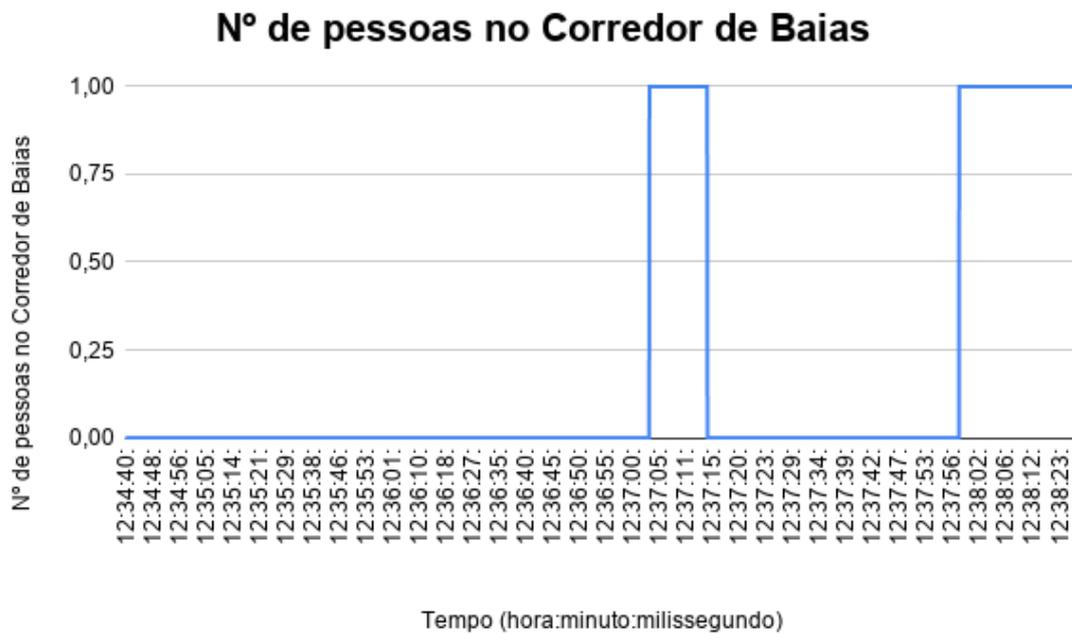


Figura 4.75: Passagem a 45 cm de distância da antena com a tag orientada perpendicularmente à antena.

Resultados para experimento com distância de 45 cm:

Número de transições reais: 10

Número de transições detectadas: 3

Precisão: 30%

Portanto, observa-se que para distâncias acima de 30 centímetros a detecção se torna muito imprecisa se a tag estiver com a face orientada perpendicularmente à antena. Desta forma, a partir desses experimentos foi constatado que a orientação da tag é fundamental na precisão da detecção das transições dos usuários entre os ambientes.

Capítulo 5

Análise dos Resultados

5.1 Resumo dos resultados

Para uma melhor análise dos resultados dos experimentos, os resultados obtidos serão resumidos em tabelas resumo, que seguem abaixo.

Resultado do experimento de comparação de performance entre leituras usando Frequência Doppler e RSSI:

Experimento de detecção utilizando efeito doppler e RSSI	
Critério de detecção	Precisão
Doppler	70,00%
RSSI	100,00%

Figura 5.1: Tabela resumo com a comparação entre as leituras usando Efeito Doppler e RSSI.

Os dados da tabela 5.1 mostram que a leitura por RSSI é muito mais confiável e preciso do que a leitura usando Efeito Doppler, chegando a ter 100% de precisão na maior parte dos casos. E por isso o RSSI foi usado como critério de passagem durante a realização do projeto.

Através dos resultados obtidos na seção "Detecção de pessoas usando informações do efeito Doppler", no capítulo 4, observou-se que nos testes feitos utilizando o Efeito Doppler como critério de detecção de passagem houveram 10 passagens reais por ambas as antenas. No entanto a antena 1 detectou em média 10.9 passagens enquanto a antena 2 detectou em média 9.4. Portanto houve um erro de 0.9% para a antena 1 e um erro de 0.4% a antena 2.

Um erro de 0.9% ou 0.4% podem parecer baixos, mas se os dados da tabela 4.1 forem analisados individualmente pode-se perceber que a variância é muito alta. No experimento N° 9 o erro chega a ser de 60%. Por causa disso o Efeito Doppler não pode ser considerado confiável para o objetivo deste projeto.

Por outro lado, através dos resultados obtidos na seção "Detecção de pessoas usando informações do RSSI", os testes feitos utilizando o RSSI como critério de detecção de passagem de tags se mostraram muito mais precisos, havendo 0% de erro. Houve 10 passagens reais e foram detectadas 10 passagens pelas antenas.

Comparando os resultados obtidos usando as duas estratégias de detecção de passagem, Efeito Doppler e RSSI, pode-se ver que enquanto a primeira é imprecisa e pouco confiável, a segunda (utilizando RSSI) é mais robusta. Por isso escolhemos esta como a principal estratégia para obtenção de dados.

Após validar o método de detecção de passagens usando o RSSI, foram feitos experimentos com o objetivo de simular movimentações em um ambiente real. Normalmente nestes ambientes haverá mais de uma pessoa transitando entre as salas. Por isso foram feitas passagens sequenciais para simular o trânsito de um grupo de pessoas entre duas salas.

Resultado dos experimentos fazendo transições sequenciais usando RSSI como critério de passagem:

Experimentos de transições sequenciais com RSSI			
Experimento	Número total de transições reais	Número de transições detectadas	Erro
1	8	8	0,00%
2	8	8	0,00%
3	8	8	0,00%
4	8	8	0,00%
5	8	7	12,50%
6	8	8	0,00%

Figura 5.2: Tabela resumo com o resultado dos experimentos com transições sequenciais.

Os dados da tabela 5.2 mostram que houve erro de detecção de passagem em apenas uma das repetições do experimento, com erro de 12,5% considerando apenas o experimento individual, e com erro aproximado de 2% considerando o total de 48 transições.

Este valor foi considerado um erro aceitável para o projeto. Por isso foi dada procedência para o próximo passo do trabalho, o acionamento da carga de acordo com a estimativa da carga térmica dentro de cada ambiente. A estimativa de carga térmica foi feita baseada no gênero de cada usuário, no entanto mais variáveis podem ser consideradas posteriormente.

Resultado dos experimentos testando o acionamento dos LEDs de acordo com o número de passagens detectadas:

Experimentos de acionamento dos leds	
Experimento	Acurácia dos acionamentos
1	100,00%
2	66,00%
3	100,00%
4	100,00%
5	66,00%
6	100,00%

Figura 5.3: Resultado do experimento dos acionamentos dos LEDs.

Os dados da tabela 5.3 mostram que em 67% dos experimentos a carga foi acionada corretamente. Através dos resultados obtidos na seção "Acionamento do ar condicionado", no capítulo 4, observando as tabelas 4.26 e 4.29, é possível observar que o erro no acionamento da carga se deve à falha de detecção da passagem das tags em alguns casos, onde a não detecção de uma passagem impede o acionamento da carga pois o valor mínimo do indicador para acionamento do relé não é alcançado. Nestes experimentos, os erros de leitura ocorrem na transição de uma tag da área externa para a sala principal, resultando no erro de acionamento da carga na sala principal. De 8 transições realizadas no experimento de número 2 e experimento de número 5, uma não foi detectada e este erro de contagem provocou o não acionamento do LED da forma desejada. Entretanto, observando todos os 6 experimentos realizados nesta seção, das 48 transições realizadas ocorreram apenas 2 erros de detecção de passagem da tag.

Percebe-se que a falha de detecção de uma passagem causa erros no acionamento da carga quando o valor mínimo da estimativa de carga térmica não é atingido. E esse erro pode ser propagado para outros ambientes quando houverem transições. A maneira encontrada para prevenir estes erros foi a adoção de um sistema não causal, onde a contagem do número de pessoas dentro de cada sala não depende da contagem na sala anterior, mas sim da detecção das transições realizadas entre os ambientes. Desta forma, caso ocorra erro em uma detecção de passagem de uma determinada tag este erro não causará problemas no sistema e pode ser corrigido automaticamente caso esta mesma tag faça outra transição entre qualquer um dos ambientes.

Em seguida foram feitos experimentos para ver a possibilidade de passagens simultâneas das tags, ou seja, quando mais de uma pessoa passa ao mesmo tempo na porta.

Resultado dos experimentos com passagens simultâneas de múltiplas tags:

Experimentos com passagens simultâneas		
Nº de tags simultâneas	Leitura	Estado do relé
2	DETECTADO	DESATIVADO
3	DETECTADO	ATIVADO
4	DETECTADO	ATIVADO
5	DETECTADO	ATIVADO

Figura 5.4: Tabela resumo do resultado dos experimentos com passagens simultâneas.

Através dos dados da tabela 5.4, que resume uma série de experimentos ilustrados pelas figuras de 4.41 a 4.56, pode-se verificar que a leitura é feita com grande precisão. Os experimentos realizados com a transição de múltiplas tags simultaneamente, onde foi analisado a detecção das passagens de todas as tags e o acionamento do relé, ocorreram da forma esperada. Como também observado nos resultados obtidos na seção "Duas passagens simultâneas em dois ambientes diferentes", no capítulo 4, percebe-se que o sistema também suporta passagens de múltiplas tags simultaneamente em ambientes diferentes. Desta forma, pode-se afirmar que o projeto supre a grande necessidade de realizar o rastreamento das tags mesmo quando ocorrem transições simultâneas, que é essencial em um ambiente com vários usuários como o LARA.

A última etapa do projeto foi realizar testes de passagens variando a distância entre a tag e a antena para verificar a precisão de detecção de passagem. Outra característica observada neste mesmo experimento foi o modo como a tag estava orientada em relação às antenas.

Os resultados dos experimentos feitos variando a distância entre a tag e a antena, com a face da tag orientada para a antena, estão ilustrados na figura 5.5.

Experimentos com a tag orientada para a antena		
Distância (cm)	Número de passagens reais	Número de passagens detectadas
30	10	10
60	10	10
90	10	9

Figura 5.5: Tabela resumo do resultado dos experimentos variando a distância entre a tag e a antena, com a tag orientada com a face para a antena.

Os resultados dos experimentos feitos variando a distância entre a tag e a antena, com a face da tag orientada perpendicularmente à antena, estão ilustrados na figura 5.6.

Experimentos com a tag orientada perpendicularmente à antena		
Distância (cm)	Número de passagens reais	Número de passagens detectadas
30	10	9
45	10	3
60	10	1
90	10	0

Figura 5.6: Tabela resumo do resultado dos experimentos variando a distância entre a tag e a antena, com a tag orientada com a face perpendicularmente à antena.

Através das tabelas 5.5 e 5.6 é possível observar que a distância entre a antena e a tag, juntamente com a orientação desta, têm grande impacto na precisão de leitura.

Quando a tag está com a face orientada para a antena é possível detectar precisamente a passagem mesmo em maiores distâncias, como a 90cm de distância. Por outro lado, quando a tag está com a face orientada perpendicularmente à antena, a precisão de leitura diminui drasticamente, chegando a ter 70% de erro quando a distância é maior do que 30cm.

Isso ocorre por causa das características da tag passiva utilizada. Como pode-se visualizar na seção "Tag Rfid Uhf Adesiva 915 - 868 Mhz Higgs3 Passiva", no capítulo 3, as tags utilizadas são adesivas e possuem uma largura muito fina. Quando a face da tag está orientada para a antena, as ondas de rádio enviadas pela leitora atingem mais facilmente a antena da tag, pois toda a área frontal, que possui uma dimensão de 96x23mm, está exposta. Por outro lado, com a tag orientada de forma perpendicular à antena, as ondas de radio emitidas não atingem a antena da tag com facilidade, pois somente a lateral da tag fica exposta, que por ser muito fina dificulta a recepção do sinal.

5.2 Visão geral dos resultados obtidos no projeto

A proposta deste projeto era realizar uma forma de monitorar em tempo real a quantidade de pessoas em um conjunto de ambientes através da tecnologia de RFID passivo de forma a utilizar as informações individuais de cada usuário presente nos ambientes para fazer uma estimativa de carga térmica e realizar o acionamento dos relés com base nessas informações.

Diante disso, de forma resumida temos que as métricas alcançadas no projeto foram:

- Precisão de leitura de passagem de tags com face orientada para antena: aproximadamente 98%;
- Média de acurácia dos acionamentos de carga: 88,7%;
- Número de passagens simultâneas precisamente detectadas: 5 tags;
- Distância máxima para boa precisão na contagem de passagens com a face da tag orientada para a antena: 90cm;
- Distância máxima para boa precisão na contagem de passagens com a face da tag orientada perpendicularmente à antena: 30cm

Com base nos dados acima, percebe-se que o monitoramento dos usuários e o acionamento das cargas ocorreram com alta precisão. Portanto, pode-se afirmar que o projeto alcançou os objetivos propostos. Entretanto, a precisão depende do modo como os equipamentos e as tags estarão disponibilizados. Com experimentos realizados, observou-se que as tags utilizadas possuem uma limitação principalmente em relação à sua orientação no momento da transição. Por conta disto, o ideal é que as tags estejam com a face orientada para a antena durante as transições, proporcionando uma melhor performance.

A eficiência deste trabalho, portanto, depende não apenas dos equipamentos e programas utilizados mas também da forma que será implementado e da sua disposição. Esses fatores estão fora do escopo do projeto.

Capítulo 6

Conclusões

Após todos os testes realizados e com base nas análises feitas no capítulo 5, foi concluído que o projeto é eficiente para monitoramento do número de pessoas no ambiente dependendo de como os equipamentos estão posicionados e da forma de usar as tags que cada pessoa carrega.

De acordo com o resultado dos experimentos, a eficiência de leitura das tags depende principalmente de três fatores: orientação da tag; distância entre tag e antena; velocidade de passagem.

A orientação da tag é o fator mais crítico na precisão de detecção de passagem de tags. Nos casos em que a tag está orientada com a face virada para a antena a precisão é bastante elevada, chegando a ser de 90% no pior caso, com distância de 90 centímetros entre a tag e a antena. Portanto, o projeto nestas condições poderia ser utilizado para contagem de pessoas em ambientes monitorados.

O pior caso, no entanto, seria onde a face da tag estivesse orientada perpendicularmente à antena. Neste caso, a precisão de detecção de passagem de tags é satisfatória apenas até uma distância de no máximo 30 centímetros entre a tag e a antena. Logo há uma redução muito alta na precisão devido à orientação da tag. O uso do projeto neste caso seria, portanto, limitada.

A distância entre a tag e a antena é o segundo fator mais importante na precisão de detecção de passagem de tags. Quanto mais longe a tag está da antena, pior será a precisão da detecção de passagem.

O último fator relevante observado neste trabalho que afeta a leitura da tag é a velocidade de passagem, caso seja superior a 18 quilômetros por hora. No entanto, em um ambiente de laboratório, como o LARA, seria improvável que alguém ultrapasse esta velocidade. Logo não vamos considerar este fator relevante.

Analisando alguns cenários, sabemos que:

Cenário 1 - Tag com face orientada para a antena: A influência da distância da tag na precisão de leitura é menor que a influência da orientação da tag. Uma tag a 90 centímetros de distância da antena com a face orientada para a antena durante a passagem obteve uma precisão de 90%.

Cenário 2 - Tag com face orientada perpendicularmente à antena: Se a tag estiver com a face orientada perpendicularmente à antena, a precisão para a mesma distância de 90 centímetros seria de 0%, não sendo detectada nenhuma passagem.

O melhor cenário seria quando a tag atravessa a passagem com a face orientada para a antena, ou seja, cenário 1. Portanto, depende do modo como o usuário está utilizando a tag.

Caso a antena estivesse na parte superior da passagem, em cima de onde o usuário passa, e a tag estivesse colada no topo da cabeça dos usuários com a face orientada para cima, as transições ocorreriam sempre no cenário 1. O resultado obtido seria o melhor possível. Entretanto, o problema disso seria a inconveniência de ter q usar a tag na cabeça, porém pode ser útil em alguns casos em que seja obrigatório a utilização de capacetes ou itens do tipo.

Caso a tag seja usada em um dos braços, a passagem ocorreria com o cenário 1 na ida e em um cenário oposto ao 1 na volta, pois em um caso a tag estaria com a face orientada para a antena e no outro caso a tag estaria com a face orientada para o lado contrário à antena, além do usuário estar entre a tag e a antena, dificultando a leitura.

Outra alternativa, a mais comum, seria o usuário usar a tag como um crachá, posicionado-o na altura do peito. Neste caso a tag atravessaria a passagem sempre com a face perpendicular à antena. Portanto, a transição seria detectada com uma precisão satisfatória se o usuário não passasse à uma distância maior que 30 centímetros da antena, o que é viável dependendo da posição da antena e do local de passagem do usuário.

Como podemos ver, para cada caso existem pontos positivos e negativos. Portanto, tudo depende da forma como o equipamento será utilizado e da finalidade para o qual será aplicado.

6.1 Perspectivas Futuras

Tendo em vista os resultados obtidos com este trabalho, alguns pontos de valor que podem ser implementados futuramente são:

- Implementação de um banco de dados como o MySQL ou Postgresql para armazenar dados dos usuários associados as tags de forma a facilitar a inserção de novas tags. Neste trabalho os dados dos usuários foram adicionados em uma lista dentro do código do programa, o que torna o processo de adicionar novos usuários difícil para quem não tem muito conhecimento do código do programa.
- Estudo mais aprofundado sobre as tags passivas, verificando se existem tags mais adequadas e que se comportem melhor. Como já foi dito, durante os experimentos se observou que a orientação da tag tem um papel fundamental nos resultados. Análises sobre parâmetros como tamanho, sensibilidade e material da tag podem ser muito úteis em trabalhos futuros.
- Implementação de um sistema supervisorio próprio para automação. Pode-se utilizar, por exemplo, o OpenHAB, que proporciona integração de diferentes sistemas de forma robusta e confiável. Dessa forma pode-se utilizá-lo em conjunto com o sistema desenvolvido neste

trabalho para realizar a automação de outros itens além do ar-condicionado, como o sistema de iluminação e integração de sensores de temperatura e umidade.

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] LAMBERTS, R. et al. Conforto e stress térmico. *LabEEE, Universidade Federal de Santa Catarina*, 2011.
- [2] CHEN, R.-S. et al. Using rfid technology in food produce traceability. *WSEAS Transactions on information science and applications*, v. 5, n. 11, p. 1551–1560, 2008.
- [3] IMPINJ. SpeedwayR Installation and Operations Guide.
- [4] LENEHAN, M. *Threshold Antenna*. Disponível em: <<https://support.impinj.com/hc/en-us/articles/202755668-Threshold-Antenna>>.
- [5] ATLASRFIDSTORE. *Using Inventory Mode in ItemTest*. Disponível em: <<https://support.atlasrfidstore.com/article/25-using-itemtest>>.
- [6] DOWNLOAD Octane SDK .NET. <https://support.impinj.com/hc/en-us/articles/360010077479-Octane-SDK-Installation-Instructions>.
- [7] OLIVEIRA, F. R. D.; ROCHA, F. S. d. P. Rfid passiva no rastreamento de usuários para automação predial. Trabalho de Graduação — Universidade de Brasília: Faculdade de Tecnologia, Brasília, DF, 2013.
- [8] ALVES, R. A.; CHUPEL, R. C. M. Utilização de rfid passiva com fusão sensorial para detecção de usuários em ambientes prediais. Trabalho de Graduação — Universidade de Brasília: Faculdade de Tecnologia, Brasília, DF, 2015.
- [9] ALMEIDA, A. A. R. d. Monitoramento de ocupação de ambientes utilizando rfid passivo. Trabalho de Graduação — Universidade de Brasília: Faculdade de Tecnologia, Brasília, DF, 2019.
- [10] RODRIGUES, A. S.; SILVA, M. P. M. *Automação de um sistema de climatização híbrido com foco no processo evaporativo visando conforto térmico e uso racional da energia*. Brasília, DF: Trabalho de Graduação — Universidade de Brasília: Faculdade de Tecnologia, 2018.
- [11] HENSEN, J. L. M. et al. *On the thermal interaction of building structure and heating and ventilating system*. [S.l.]: Technische Universiteit Eindhoven, 1991.
- [12] ASHRAE. *Standard 55 Thermal Environment Conditions for Human Occupancy*. [S.l.: s.n.], 2004.

- [13] VIANNA, P. I. E. *Automação de um sistema de climatização híbrido com foco no processo evaporativo visando conforto térmico e uso racional da energia*. [S.l.]: Trabalho de Graduação — Universidade de Brasília: Faculdade de Tecnologia, 2012.
- [14] FANGER, P. O. et al. Thermal comfort. analysis and applications in environmental engineering. *Thermal comfort. Analysis and applications in environmental engineering.*, Copenhagen: Danish Technical Press., 1970.
- [15] ARAÚJO, A. M. *Controle TITO via desacoplamento de malhas de sistema de climatização híbrido*. Brasília, DF: Trabalho de Graduação — Universidade de Brasília: Faculdade de Tecnologia, 2016.
- [16] OLIVEIRA, F. A. R. *Instrumentação e identificação de um ambiente predial visando controle preditivo do conforto térmico*. [S.l.]: Dissertação de Mestrado em Engenharia Elétrica — Universidade de Brasília: Faculdade de Tecnologia, 2010.
- [17] CREDER, H. *Instalações de ar condicionado*. [S.l.]: Livros Tecnicos e Cientificos, 2004.
- [18] RFID Middleware. <http://www.oxxcode.com.br/middleware-rfid/>.
- [19] RSSI Role in RFID. <https://www.atlasrfidstore.com/rfid-insider/rssi-role-rfid>.
- [20] VISUAL Studio. <https://docs.microsoft.com/pt-br/visualstudio/get-started/visual-studio-ide?view=vs-2019>.
- [21] APLICATIVO Windows Forms. <https://docs.microsoft.com/en-us/dotnet/desktop/winforms/overview/?view=netdesktop-5.0>.
- [22] DJONGYANG, N.; TCHINDA, R.; NJOMO, D. Thermal comfort: A review paper. *Renewable and sustainable energy reviews*, Elsevier, v. 14, n. 9, p. 2626–2640, 2010.
- [23] FROIZ-MÍGUEZ, I. et al. Design, implementation and practical evaluation of an iot home automation system for fog computing applications based on mqtt and zigbee-wifi sensor nodes. *Sensors*, Multidisciplinary Digital Publishing Institute, v. 18, n. 8, p. 2660, 2018.
- [24] BHATT, A.; PATOLIYA, J. Cost effective digitization of home appliances for home automation with low-power wifi devices. In: IEEE. *2016 2nd International Conference on Advances in Electrical, Electronics, Information, Communication and Bio-Informatics (AEEICB)*. [S.l.], 2016. p. 643–648.
- [25] ATZORI, L.; IERA, A.; MORABITO, G. The internet of things: A survey. *Computer networks*, Elsevier, v. 54, n. 15, p. 2787–2805, 2010.
- [26] GUBBI, J. et al. Internet of things (iot): A vision, architectural elements, and future directions. *Future generation computer systems*, Elsevier, v. 29, n. 7, p. 1645–1660, 2013.
- [27] GLUHAK, A. et al. A survey on facilities for experimental internet of things research. 2011.
- [28] ASHTON, K. et al. That ‘internet of things’ thing. *RFID journal*, v. 22, n. 7, p. 97–114, 2009.

- [29] LACERDA, F. *Arquitetura da informação pervasiva: projetos de ecossistemas de informação na internet das coisas*. 2015.
- [30] SAS: The power to know. https://www.sas.com/pt_br/insights/analytics/computer-vision.html. Acessado em: 01/07/2019.
- [31] MOLLOY, D. *Exploring raspberry PI*. [S.l.]: John Wiley and Sons, 2016.
- [32] RASPBERRY Pi 3 Model B. <https://www.raspberrypi.org/products/raspberry-pi-3-model-b/>. Acessado em: 15/05/2019.
- [33] GPIO: RASPBERRY PI MODELS A AND B. <https://www.raspberrypi.org/documentation/usage/gpio/>. Acessado em: 15/05/2019.
- [34] ATLASRFIDSTORE. *Impinj Speedway Revolution R420 UHF RFID Reader (4 Port)*. Disponível em: <<https://www.atlasrfidstore.com/impinj-speedway-revolution-r420-uhf-rfid-reader-4-port/>>.
- [35] NEXTPOINTS. *IMPINJ THRESHOLD NPA PASSLINE UHF RFID ANTENNA*. Disponível em: <<https://nextpoints.com/en/rfid-products/rfid-antenna/impinj-threshold-rfid-antenna/>>.
- [36] LENEHAN, M. *Performing a Margin Test using Impinj ItemTest software*. Disponível em: <<https://support.impinj.com/hc/en-us/articles/202756388-Performing-a-Margin-Test-using-Impinj-ItemTest-software>>.
- [37] IMPACTOS da melhoria na eficiência de ar-condicionado. https://iei-brasil.org/wp-content/uploads/2019/02/Relatorio_Kigali.pdf.
- [38] AR-CONDICIONADO aumenta conta de luz em até 50%. <https://www.uol.com.br/tilt/noticias/redacao/2019/01/21/ar-condicionado-aumenta-conta-de-luz-em-ate-50-veja-como-economizar.htm#:~:text=A%20raz%C3%A3o%20para%20isso%20%C3%A9,apenas%20uma%20hora%20por%20dia.>
- [39] APLICATIVO Windows Forms 2. <https://docs.microsoft.com/pt-br/dotnet/desktop/winforms/windows-forms-overview?view=netframeworkdesktop-4.8>.

ANEXOS

I. PROGRAMAS UTILIZADOS

Neste anexo se encontram os programas desenvolvidos para este trabalho. Utilizando o Visual Studio, Windows Forms e o kit de desenvolvimento Octane SDK disponibilizado pela Impinj, foram feitos os seguintes programas realizados utilizando a linguagem C#:

- MainProgram.cs
- VariaveisProgram.cs
- Transicao.cs
- SerialCom.cs
- Program.cs
- Form1.designer.cs

Nas páginas seguintes estarão os códigos referentes aos programas feitos para a realização deste trabalho.

MainProgram.cs

```
1 ////////////////////////////////////////////////////////////////////
2 //
3 // Trabalho de Graduação 2 – Leonardo Pereira e Kenneth Lui
4 // Universidade de Brasília – UnB
5 // Novembro de 2020
6 //
7 ////////////////////////////////////////////////////////////////////
8
9 /* Aviso: A biblioteca 'Octane SDK' foi desenvolvida e é propriedade da
10 empresa IMPINJ, INC., e está sendo utilizada com base na licença de software
11 aberto descrita a seguir
12 *
13 * Para mais informações, visitar o site: https://support.impinj.com/hc/en-us/articles/360000468370-Software-Tools-License-Disclaimer
14 */
15
16 /*PLEASE READ THE FOLLOWING LICENSE & DISCLAIMER (AGREEMENT) CAREFULLY
17 BEFORE USING ANY SOFTWARE TOOLS (AS DEFINED BELOW) MADE AVAILABLE TO YOU
18 (LICENSEE) BY IMPINJ, INC. (IMPINJ). BY USING THE SOFTWARE TOOLS, YOU
19 ACKNOWLEDGE THAT YOU HAVE READ AND UNDERSTOOD ALL THE TERMS AND CONDITIONS
20 OF THE AGREEMENT, YOU WILL BE CONSENTING TO BE BOUND BY THEM, AND YOU ARE
21 AUTHORIZED TO DO SO. IF YOU DO NOT ACCEPT THESE TERMS AND CONDITIONS, DO NOT
22 USE THE SOFTWARE TOOLS.
23
24 1. PURPOSE OF AGREEMENT. From time to time, Impinj technical personnel may
25 make available to Licensee certain software, including code (in source and
26 object form), tools, libraries, configuration files, translations, and
27 related documentation (collectively, Software Tools), upon specific
28 request or to assist with a specific deployment. This Agreement sets forth
29 Licensee's limited rights and Impinj's limited obligations with respect to
30 the Software Tools. Licensee acknowledges that Impinj provides the Software
31 Tools free of charge. This Agreement does not grant any rights with respect
32 to Impinj standalone software products (e.g., ItemSense, ItemEncode,
33 SpeedwayConnect) or the firmware on Impinj hardware, all of which are
34 subject to separate license terms.
35
36 2. LIMITED LICENSE. Subject to the terms and conditions of this Agreement,
37 Impinj hereby grants to Licensee a limited, royalty-free, worldwide, nonexclusive,
38 perpetual and irrevocable (except as set forth below), nontransferable license,
39 without right of sublicense, to (a) use the Software
40 Tools and (b) only with respect to Software Tools provided in source code
41 form, modify and create derivative works of such Software Tools, in each
42 case, solely for Licensee's internal development related to the deployment
43 of Impinj products (Purpose). The Software Tools may only be used by
44 employees of Licensee that must have access to the Software Tools in
45 connection with the Purpose.
46
47 3. TERMINATION. Impinj may immediately terminate this Agreement if Licensee
48 breaches any provision hereof. Upon the termination of this Agreement,
49 Licensee must (a) discontinue all use of the Software Tools, (b) uninstall
50 the Software Tools from its systems, (c) destroy or return to Impinj all
51 copies of the Software Tools and any other materials provided by Impinj, and
52 (d) promptly provide Impinj with written confirmation (including via email)
53 of Licensee's compliance with these provisions. Sections 4–10 will survive
54 termination of this Agreement.
55
56 4. OWNERSHIP. The Software Tools are licensed, not sold, by Impinj to
57 Licensee. Impinj and its suppliers own and retain all right, title, and
58 interest, including all intellectual property rights, in and to the Software
59 Tools. Except for those rights expressly granted in this Agreement, no
60 other rights are granted, either express or implied, to Licensee. Impinj
61 reserves the right to develop, price and sell software products that have
62 features similar to or competitive with Software Tools. Licensee grants
63 Impinj a limited, royalty-free, worldwide, perpetual and irrevocable,
64 transferable, sublicensable, license to Licensee's derivative works of
65 Software Tools; provided that Licensee has no obligation under this
66 Agreement to deliver to Impinj any such derivative works.
67
68 5. CONFIDENTIALITY. In order to protect the trade secrets and proprietary
69 know-how contained in the Software Tools, Licensee will not decompile,
70 disassemble, or reverse engineer, or otherwise attempt to gain access to the
71 source code or algorithms of the Software Tools (unless Impinj provides the
72 Software Tools in source code format). Licensee will maintain the
73 confidentiality of and not disclose to any third party: (a) all non-public
74 information disclosed by Impinj to Licensee under this Agreement and (b) all
75 performance data and all other information obtained through the Software
76 Tools.
77
78 6. WARRANTY DISCLAIMER. LICENSEE ACKNOWLEDGES THAT IMPINJ PROVIDES THE
79 SOFTWARE TOOLS FREE OF CHARGE AND ONLY FOR THE PURPOSE. ACCORDINGLY, THE
```

80 SOFTWARE TOOLS ARE PROVIDED AS IS WITHOUT QUALITY CHECK, AND IMPINJ DOES
81 NOT WARRANT THAT THE SOFTWARE TOOLS WILL OPERATE WITHOUT ERROR OR
82 INTERRUPTION OR MEET ANY PERFORMANCE STANDARD OR OTHER EXPECTATION. IMPINJ
83 EXPRESSLY DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING THE
84 IMPLIED WARRANTIES OF MERCHANTABILITY, NONINFRINGEMENT, QUALITY, ACCURACY,
85 AND FITNESS FOR A PARTICULAR PURPOSE. IMPINJ IS NOT OBLIGATED IN ANY WAY TO
86 PROVIDE SUPPORT OR OTHER MAINTENANCE WITH RESPECT TO THE SOFTWARE TOOLS.
87

88 7. LIMITATION OF LIABILITY. THE TOTAL LIABILITY OF IMPINJ ARISING OUT OF OR
89 RELATED TO THE SOFTWARE TOOLS WILL NOT EXCEED THE TOTAL AMOUNT PAID BY
90 LICENSEE TO IMPINJ PURSUANT TO THIS AGREEMENT. IN NO EVENT WILL IMPINJ HAVE
91 LIABILITY FOR ANY INDIRECT, INCIDENTAL, SPECIAL, OR CONSEQUENTIAL DAMAGES,
92 EVEN IF ADVISED OF THE POSSIBILITY OF THESE DAMAGES. THESE LIMITATIONS WILL
93 APPLY NOTWITHSTANDING ANY FAILURE OF ESSENTIAL PURPOSE OF ANY LIMITED REMEDY
94 IN THIS AGREEMENT.
95

96 8. THIRD PARTY SOFTWARE. The Software Tools may contain software created by a
97 third party. Licensee's use of any such third party software is subject to
98 the applicable license terms and this Agreement does not alter those license
99 terms. Licensee may not subject any portion of the Software Tools to an
100 open source license.
101

102 9. RESTRICTED USE. Licensee will comply with all applicable laws and
103 regulations to preclude the acquisition by any governmental agency of
104 unlimited rights to technical data, software, and documentation provided
105 with Software Tools, and include the appropriate Restricted Rights or
106 Limited Rights notices required by the applicable U.S. or foreign
107 government agencies. Licensee will comply in all respects with all U.S. and
108 foreign export and re-export laws and regulations applicable to the
109 technology and documentation provided hereunder.
110

111 10. MISCELLANEOUS. This Agreement will be governed by the laws of the State of
112 Washington, U.S.A without reference to conflict of law principles. All
113 disputes arising out of or related to it, will be subject to the exclusive
114 jurisdiction of the state and federal courts located in King County,
115 Washington, and the parties agree and submit to the personal and exclusive
116 jurisdiction and venue of these courts. Licensee will not assign this
117 Agreement, directly or indirectly, by operation of law or otherwise, without
118 the prior written consent of Impinj. This Agreement (and any applicable
119 nondisclosure agreement) is the entire agreement between the parties
120 relating to the Software Tools. No waiver or modification of this Agreement
121 will be valid unless contained in a writing signed by each party.
122 */
123

```
124
125
126 using System;
127 using Impinj.OctaneSdk;
128 using System.Collections;
129 using System.Collections.Generic;
130 using System.Text;
131 using System.IO;
132 using System.Threading;
133 using System.Security.Cryptography.X509Certificates;
134 using TeGe;
135 using System.Windows.Forms;
136
137 namespace TeGe2
138 {
139
140     // Programa principal.
141     class Program
142     {
143         // Create an instance of the ImpinjReader class.
144         static ImpinjReader reader = new ImpinjReader();
145
146         protected static List<ImpinjReader> readers = new List<ImpinjReader>();
147
148         static void Main(string[] args)
149         {
150             try
151             {
152                 // Ao iniciar o programa, seta o fileHandler e cria
153                 // os arquivos que armazenam os dados da leitura.
154                 Console.WriteLine("Iniciando programa...");
155                 GlobalData.filehandler.SetFileHandler();
156                 GlobalData.filehandler.CreateFile();
157
158                 Thread.Sleep(1000);
159
160                 // Hostname de todas as leitores utilizadas.
```

```

161 string hostname1 = "SpeedwayR-10-9F-3F.local";
162 string hostname2 = "SpeedwayR-10-9F-C8.local";
163 string hostname3 = "SpeedwayR-10-9F-BB.local";
164
165 // Adicionando todas as leitoras à lista.
166 readers.Add(new ImpinjReader(hostname1, "Reader #1"));
167 readers.Add(new ImpinjReader(hostname2, "Reader #2"));
168 readers.Add(new ImpinjReader(hostname3, "Reader #3"));
169
170 // Faz o reset dos leds.
171 SerialCom.Controle_Ar("Sala Principal"); Thread.Sleep(500);
172 SerialCom.Controle_Ar("Sala Reuniões"); Thread.Sleep(500);
173 SerialCom.Controle_Ar("Corredor de Baias"); Thread.Sleep(500);
174
175 // Conecta os readers e seta as configuracoes necessarias de cada uma das leitoras.
176 foreach (ImpinjReader reader in readers)
177 {
178     // Conecta com o reader.
179     // Troque o ReaderHostname em ConstantesReader.cs para o endereço IP ou hostname do seu reader.
180     //reader.Connect(ConstantesReader.ReaderHostname);
181     reader.Connect();
182
183     // Get the default settings
184     // We'll use these as a starting point and then modify the settings we're interested in.
185     Settings settings = reader.QueryDefaultSettings();
186
187     // Tell the reader to include the RF doppler frequency in all tag reports.
188     settings.Report.IncludeDopplerFrequency = true;
189     settings.Report.IncludePeakRssi = true;
190     settings.Report.IncludeAntennaPortNumber = true;
191     settings.Report.IncludeLastSeenTime = true;
192
193     // Use antenna #1 and #2.
194     settings.Antennas.DisableAll();
195     settings.Antennas.GetAntenna(1).IsEnabled = true;
196     settings.Antennas.GetAntenna(2).IsEnabled = true;
197
198     //settings.Antennas.TxPowerMax = true;
199     //settings.Antennas.RxSensitivityMax = true;
200     //settings.Antennas.TxPowerInDbm = 20.0;
201     //settings.Antennas.RxSensitivityInDbm = -70.0;
202
203     // ReaderMode must be set to DenseReaderM8.
204     settings.ReaderMode = ReaderMode.DenseReaderM8;
205
206     // Apply the newly modified settings.
207     reader.ApplySettings(settings);
208
209     // Assign the TagsReported event handler.
210     // This specifies which method to call
211     // when tags reports are available.
212     reader.TagsReported += AnalisaDadosTAG;
213
214     // Start reading.
215     reader.Start();
216 }
217
218 // Inicia a interface grafica
219 Thread GUI = new Thread(() => ProgramGUI.MainGUI());
220 GUI.Start();
221
222 // Wait for the user to press enter.
223 Console.WriteLine("Programa Iniciado.\nPressione Enter para sair do programa.\n");
224 Console.ReadLine();
225
226 // Para de ler e encerra a conexao com os readers
227 foreach (ImpinjReader reader in readers)
228 {
229     // Stop reading.
230     reader.Stop();
231
232     // Disconnect from the reader.
233     reader.Disconnect();
234 }
235
236 }
237 catch (OctaneSdkException e)
238 {
239     // Handle Octane SDK errors.
240     Console.WriteLine("Octane SDK exception: {0}", e.Message);
241 }

```

```

242     catch (Exception e)
243     {
244         // Handle other .NET errors.
245         Console.WriteLine("Exception : {0}", e.Message);
246     }
247 }
248
249
250
251 // Funcao que faz a analise dos reports das TAGs dado uma determinada leitora.
252 private static void AnalisaDadosTAG(ImpinjReader sender, TagReport report)
253 {
254     switch (sender.Address)
255     {
256         // Caso o report da TAG seja relacionado a Leitora #1, entra aqui.
257         case "SpeedwayR-10-9F-3F.local":
258             foreach (Tag tag in report)
259             {
260
261                 // Variavel do tipo Tags_TG que vai armazenar a tag que foi reportada agora.
262                 Tags_TG TagAtual;
263
264                 // Variavel que indica se aceita o report da TAG ou nao.
265                 int TAGsReportsON = 0;
266
267                 // Variaveis usadas no foreach para determinar o indice da TAG na lista de tags.
268                 int contador = 0;
269                 int indice = 0;
270
271                 // Verifica o EPC das tags que estao registradas para verificar se o report dessa TAG
272                 // deve ser aceito ou nao dependendo da variavel ReportON da TAG. Caso seja true,
273                 // a variavel TAGReportON eh setada e o report eh aceito.
274                 foreach (var TagRegistrada in GlobalData.ListaTAGs)
275                 {
276                     if (TagRegistrada.EPC == tag.Epc.ToString())
277                     {
278                         {
279                             indice = contador;
280                             TagAtual = TagRegistrada;
281                             if (TagAtual.ReportON == true)
282                             {
283                                 TAGsReportsON = 1;
284                             }
285                         }
286                         contador++;
287                     }
288
289                     // Condicional com um filtro para validar leitura. Valor de RSSI tem que estar acima
290                     // do valor do filtro e TAGsReportsON tem que ser verdadeiro.
291                     if (tag.PeakRssiInDbm > GlobalData.FiltroPassaBaixaRSSI && TAGsReportsON == 1)
292                     {
293                         // Checa se a antena 1 que está lendo
294                         if (tag.AntennaPortNumber == 1)
295                         {
296                             // Se o RSSI lido do report for maior que o pico atual
297                             // de RSSI, atualiza o valor máximo de RSSI e seu tempo.
298                             if (tag.PeakRssiInDbm > GlobalData.ListaTAGs[indice].Leitora1.PicoRSSIAnt1 && GlobalData.ListaTAGs[
299                                 ↵ indice].Leitora1.LockPicoRSSIAnt1 == 0)
300                             {
301                                 GlobalData.ListaTAGs[indice].Leitora1.PicoRSSIAnt1 = tag.PeakRssiInDbm;
302                                 GlobalData.ListaTAGs[indice].Leitora1.TempoPicoRSSIAnt1 = DateTime.Now.ToString("HH-mm-ss-fff"
303                                     ↵ );
304                                 GlobalData.ListaTAGs[indice].Leitora1.DateTimePicoRSSIAnt1 = DateTime.Now;
305
306                                 // Verifica se o pico de RSSI da antena 1 é maior ou igual que o indicador RSSI de passagem.
307                                 // Se for, atualiza a flag que indica que houve passagem pela primeira antena e
308                                 // da "lock" nesse valor, para que nao pegue outras medidas desnecessariamente.
309                                 if (GlobalData.ListaTAGs[indice].Leitora1.PicoRSSIAnt1 >= GlobalData.IndicadorRSSIPassagem)
310                                 {
311                                     GlobalData.ListaTAGs[indice].Leitora1.FlagPassagemAnt1 = 1;
312                                     GlobalData.ListaTAGs[indice].Leitora1.LockPicoRSSIAnt1 = 1;
313                                     LeitoraUm.ContadorPassagemAnt1 += 1;
314
315                                     // Faz o reset das variaveis da TAG caso passe 5 segundos desde um lock e não ocorra uma passagem
316                                     if (GlobalData.ListaTAGs[indice].Leitora1.LockPicoRSSIAnt2 == 0)
317                                     {
318                                         GlobalData.ListaTAGs[indice].TimerReset = 1;
319                                         Thread TimerResetTAG = new Thread(() => MonitoramentoTAG.TimerResetTAG(indice));
320                                         TimerResetTAG.Start();
321                                     }
322                                 }
323                             }
324                         }
325                     }
326                 }
327             }
328         }
329     }
330 }

```

```

321     }
322
323
324     // Checa se a antena 2 que está lendo.
325     if (tag.AntennaPortNumber == 2)
326     {
327         // Se o RSSI lido do report for maior que o pico atual
328         // de RSSI, atualiza o valor máximo de RSSI e seu tempo.
329         if (tag.PeakRssiInDbm > GlobalData.ListaTAGs[indice].Leitora1.PicoRSSIAnt2 && GlobalData.ListaTAGs[
            ↪ indice].Leitora1.LockPicoRSSIAnt2 == 0)
330         {
331             GlobalData.ListaTAGs[indice].Leitora1.PicoRSSIAnt2 = tag.PeakRssiInDbm;
332             GlobalData.ListaTAGs[indice].Leitora1.TempoPicoRSSIAnt2 = DateTime.Now.ToString("HH-mm-ss-fff"
            ↪ );
333             GlobalData.ListaTAGs[indice].Leitora1.DateTimePicoRSSIAnt2 = DateTime.Now;
334
335             // Verifica se o pico de RSSI da antena 1 é maior ou igual que o indicador RSSI de passagem.
336             // Se for, atualiza a flag que indica que houve passagem pela primeira antena e
337             // da "lock" nesse valor, para que não pegue outras medidas desnecessariamente.
338             if (GlobalData.ListaTAGs[indice].Leitora1.PicoRSSIAnt2 >= GlobalData.IndicadorRSSIPassagem)
339             {
340                 GlobalData.ListaTAGs[indice].Leitora1.FlagPassagemAnt2 = 1;
341                 GlobalData.ListaTAGs[indice].Leitora1.LockPicoRSSIAnt2 = 1;
342                 LeitoraUm.ContadorPassagemAnt2 += 1;
343
344                 // Faz o reset das variáveis da TAG caso passe 5 segundos desde um lock e não ocorra uma passagem
345                 if (GlobalData.ListaTAGs[indice].Leitora1.LockPicoRSSIAnt1 == 0)
346                 {
347                     GlobalData.ListaTAGs[indice].TimerReset = 1;
348                     Thread TimerResetTAG = new Thread(() => MonitoramentoTAG.TimerResetTAG(indice));
349                     TimerResetTAG.Start();
350                 }
351             }
352         }
353     }
354
355     // Checa se a flag de passagem das duas antenas foram setadas.
356     // Se sim, significa que houve a passagem da TAG pelas duas antenas.
357     // Ou seja, significa que alguém entrou ou saiu do ambiente.
358     if (GlobalData.ListaTAGs[indice].Leitora1.FlagPassagemAnt1 == 1 && GlobalData.ListaTAGs[indice].Leitora1.
            ↪ FlagPassagemAnt2 == 1)
359     {
360         // Checa os tempos de cada pico para saber se a TAG entrou ou saiu do ambiente.
361         // Se a comparação for negativa, significa que passaram primeiro pela antena 1 e depois
362         // pela antena 2, ou seja, a pessoa entrou no ambiente.
363         if (DateTime.Compare(GlobalData.ListaTAGs[indice].Leitora1.DateTimePicoRSSIAnt1, GlobalData.
            ↪ ListaTAGs[indice].Leitora1.DateTimePicoRSSIAnt2) < 0)
364         {
365             // Como acabou de ocorrer uma transição, seta a variável ReportON para falso para que
366             // essa tag não receba reports por alguns segundos, evitando erro nas análises futuras.
367             GlobalData.ListaTAGs[indice].ReportON = false;
368             bool passagem = true;
369
370             Console.WriteLine("\n\nLeitora endereço: {0}", sender.Address);
371
372             // Verifica o EPC das tags que estão registradas para verificar quem entrou no ambiente,
373             // e aí passa o nome da pessoa para a função MonitoramentoSALA.
374             TransicaoTags.transicao_sala(sender.Address, GlobalData.ListaTAGs[indice].EPC, passagem, GlobalData.
            ↪ ListaTAGs[indice].Nome, GlobalData.ListaTAGs[indice]);
375
376             // Seta a flag que indica que está ocorrendo uma transição entre ambientes.
377             //LeitoraUm.FlagTrocaAmbiente = 1;
378             //GlobalData.ListaTAGs[indice].Leitora1.FlagTrocaAmbiente = 1;
379             MonitoramentoTAG.ResetaTAG(1, indice);
380
381             // Inicializa a Thread que faz o descarte de reports das tags por 2 segundos.
382             // Isso é feito para que os dados dos reports durante a transição da pessoa
383             // não atrapalhem a análise.
384             Thread DescartaReportThread = new Thread(() => MonitoramentoTAG.DescartaReport(indice));
385             DescartaReportThread.Start();
386         }
387
388         // Checa os tempos de cada pico para saber se a TAG entrou ou saiu do ambiente.
389         // Se a comparação for positiva, significa que passaram primeiro pela antena 2 e depois
390         // pela antena 1, ou seja, a pessoa saiu do ambiente.
391         if (DateTime.Compare(GlobalData.ListaTAGs[indice].Leitora1.DateTimePicoRSSIAnt1, GlobalData.
            ↪ ListaTAGs[indice].Leitora1.DateTimePicoRSSIAnt2) > 0)
392         {
393             // Como acabou de ocorrer uma transição, seta a variável ReportON para falso para que
394             // essa tag não receba reports por alguns segundos, evitando erro nas análises futuras.
395             GlobalData.ListaTAGs[indice].ReportON = false;

```

```

396         bool passagem = false;
397
398         Console.WriteLine("\n\nLeitora endereço: {0}", sender.Address);
399
400         // Verifica o EPC das tags que estao registradas para verificar quem saiu do ambiente,
401         // e ai passa o nome da pessoa para a funcao MonitoramentoSALA.
402         TransicaoTags.transicao_sala(sender.Address, GlobalData.ListaTAGs[indice].EPC, passagem, GlobalData.
            ↳ ListaTAGs[indice].Nome, GlobalData.ListaTAGs[indice]);
403
404         // Seta a flag que indica que esta ocorrendo uma transicao entre ambientes.
405         //LeitoraUm.FlagTrocaAmbiente = 1;
406         //GlobalData.ListaTAGs[indice].Leitora1.FlagTrocaAmbiente = 1;
407         MonitoramentoTAG.ResetaTAG(1, indice);
408
409         // Inicializa a Thread que faz o descarte de reports das tags por 2 segundos.
410         // Isso eh feito para que os dados dos reports durante a transicao da pessoa
411         // nao atrapalhem a analise.
412         Thread DescartaReportThread = new Thread(() => MonitoramentoTAG.DescartaReport(indice));
413         DescartaReportThread.Start();
414     }
415 }
416
417 // Armazena o DateTime do tempo da ultima vez que foi feito uma leitura valida.
418 GlobalData.LastSeen = DateTime.Now;
419
420 //Escreve nos arquivos os dados.
421 if (tag.AntennaPortNumber == 1)
422 {
423     GlobalData.filehandler.WriteToFile(tag.Epc.ToString(), sender.Name, tag.AntennaPortNumber, tag.
            ↳ RfDopplerFrequency.ToString("0.00"), tag.PeakRssiInDbm.ToString(), "", GlobalData.
            ↳ DictSalaReunioes.Count, GlobalData.DictSalaPrincipal.Count, GlobalData.DictCorredorBaias.Count
            ↳ , GlobalData.DictAmbienteExterno.Count, GlobalData.AcionamtSalaReunioes, GlobalData.
            ↳ AcionamtSalaPrincipal, GlobalData.AcionamtCorredorBaias);
424 }
425 else
426 {
427     GlobalData.filehandler.WriteToFile(tag.Epc.ToString(), sender.Name, tag.AntennaPortNumber, tag.
            ↳ RfDopplerFrequency.ToString("0.00"), "", tag.PeakRssiInDbm.ToString(), GlobalData.
            ↳ DictSalaReunioes.Count, GlobalData.DictSalaPrincipal.Count, GlobalData.DictCorredorBaias.Count
            ↳ , GlobalData.DictAmbienteExterno.Count, GlobalData.AcionamtSalaReunioes, GlobalData.
            ↳ AcionamtSalaPrincipal, GlobalData.AcionamtCorredorBaias);
428 }
429 }
430 }
431 break;
432
433 // Caso o report da TAG seja relacionado a Leitora #2, entra aqui.
434 case "SpeedwayR-10-9F-C8.local":
435     foreach (Tag tag in report)
436     {
437
438         // Variavel do tipo Tags_TG que vai armazenar a tag que foi reportada agora.
439         Tags_TG TagAtual;
440
441         // Variavel que indica se aceita o report da TAG ou nao.
442         int TAGsReportsON = 0;
443
444         // Variaveis usadas no foreach para determinar o indice da TAG na lista de tags.
445         int contador = 0;
446         int indice = 0;
447
448         // Verifica o EPC das tags que estao registradas para verificar se o report dessa TAG
449         // deve ser aceito ou nao dependendo da variavel ReportON da TAG. Caso seja true,
450         // a variavel TAGReportON eh setada e o report eh aceito.
451         foreach (var TagRegistrada in GlobalData.ListaTAGs)
452         {
453             if (TagRegistrada.EPC == tag.Epc.ToString())
454             {
455                 TagAtual = TagRegistrada;
456                 indice = contador;
457                 if (TagAtual.ReportON == true)
458                 {
459                     TAGsReportsON = 1;
460                 }
461             }
462             contador++;
463         }
464
465         // Condicional com um filtro para validar leitura. Valor de RSSI tem que estar acima
466         // do valor do filtro e TAGsReportsON tem que ser verdadeiro.
467         if (tag.PeakRssiInDbm > GlobalData.FiltroPassaBaixaRSSI && TAGsReportsON == 1)

```

```

468     {
469         // Checa se a antena 1 que está lendo
470         if (tag.AntennaPortNumber == 1)
471         {
472             // Se o RSSI lido do report for maior que o pico atual
473             // de RSSI, atualiza o valor máximo de RSSI e seu tempo.
474             if (tag.PeakRssiInDbm > GlobalData.ListaTAGs[indice].Leitora2.PicoRSSIAnt1 && GlobalData.ListaTAGs[
475                 ↪ indice].Leitora2.LockPicoRSSIAnt1 == 0)
476             {
477                 GlobalData.ListaTAGs[indice].Leitora2.PicoRSSIAnt1 = tag.PeakRssiInDbm;
478                 GlobalData.ListaTAGs[indice].Leitora2.TempoPicoRSSIAnt1 = DateTime.Now.ToString("HH-mm-ss-fff"
479                 ↪ );
480                 GlobalData.ListaTAGs[indice].Leitora2.DateTimePicoRSSIAnt1 = DateTime.Now;
481
482                 // Verifica se o pico de RSSI da antena 1 é maior ou igual que o indicador RSSI de passagem.
483                 // Se for, atualiza o flag que indica que houve passagem pela primeira antena e
484                 // da "lock" nesse valor, para que não pegue outras medidas desnecessariamente.
485                 if (GlobalData.ListaTAGs[indice].Leitora2.PicoRSSIAnt1 >= GlobalData.IndicadorRSSIPassagem)
486                 {
487                     GlobalData.ListaTAGs[indice].Leitora2.FlagPassagemAnt1 = 1;
488                     GlobalData.ListaTAGs[indice].Leitora2.LockPicoRSSIAnt1 = 1;
489                     LeitoraDois.ContadorPassagemAnt1 += 1;
490
491                     // Faz o reset das variáveis da TAG caso passe 5 segundos desde um lock e não ocorra uma passagem
492                     if (GlobalData.ListaTAGs[indice].Leitora2.LockPicoRSSIAnt2 == 0)
493                     {
494                         GlobalData.ListaTAGs[indice].TimerReset = 2;
495                         Thread TimerResetTAG = new Thread(() => MonitoramentoTAG.TimerResetTAG(indice));
496                         TimerResetTAG.Start();
497                     }
498                 }
499             }
500         }
501         // Checa se a antena 2 que está lendo.
502         if (tag.AntennaPortNumber == 2)
503         {
504             // Se o RSSI lido do report for maior que o pico atual
505             // de RSSI, atualiza o valor máximo de RSSI e seu tempo.
506             if (tag.PeakRssiInDbm > GlobalData.ListaTAGs[indice].Leitora2.PicoRSSIAnt2 && GlobalData.ListaTAGs[
507                 ↪ indice].Leitora2.LockPicoRSSIAnt2 == 0)
508             {
509                 GlobalData.ListaTAGs[indice].Leitora2.PicoRSSIAnt2 = tag.PeakRssiInDbm;
510                 GlobalData.ListaTAGs[indice].Leitora2.TempoPicoRSSIAnt2 = DateTime.Now.ToString("HH-mm-ss-fff"
511                 ↪ );
512                 GlobalData.ListaTAGs[indice].Leitora2.DateTimePicoRSSIAnt2 = DateTime.Now;
513
514                 // Verifica se o pico de RSSI da antena 1 é maior ou igual que o indicador RSSI de passagem.
515                 // Se for, atualiza o flag que indica que houve passagem pela primeira antena e
516                 // da "lock" nesse valor, para que não pegue outras medidas desnecessariamente.
517                 if (GlobalData.ListaTAGs[indice].Leitora2.PicoRSSIAnt2 >= GlobalData.IndicadorRSSIPassagem)
518                 {
519                     GlobalData.ListaTAGs[indice].Leitora2.FlagPassagemAnt2 = 1;
520                     GlobalData.ListaTAGs[indice].Leitora2.LockPicoRSSIAnt2 = 1;
521                     LeitoraDois.ContadorPassagemAnt2 += 1;
522
523                     // Faz o reset das variáveis da TAG caso passe 5 segundos desde um lock e não ocorra uma passagem
524                     if (GlobalData.ListaTAGs[indice].Leitora2.LockPicoRSSIAnt1 == 0)
525                     {
526                         GlobalData.ListaTAGs[indice].TimerReset = 2;
527                         Thread TimerResetTAG = new Thread(() => MonitoramentoTAG.TimerResetTAG(indice));
528                         TimerResetTAG.Start();
529                     }
530                 }
531             }
532         }
533         // Checa se a flag de passagem das duas antenas foram setadas.
534         // Se sim, significa que houve a passagem da TAG pelas duas antenas.
535         // Ou seja, significa que alguém entrou ou saiu do ambiente.
536         if (GlobalData.ListaTAGs[indice].Leitora2.FlagPassagemAnt1 == 1 && GlobalData.ListaTAGs[indice].Leitora2.
537             ↪ FlagPassagemAnt2 == 1)
538         {
539             // Checa os tempos de cada pico para saber se a TAG entrou ou saiu do ambiente.
540             // Se a comparação for negativa, significa que passaram primeiro pela antena 1 e depois
541             // pela antena 2, ou seja, a pessoa entrou no ambiente.
542             if (DateTime.Compare(GlobalData.ListaTAGs[indice].Leitora2.DateTimePicoRSSIAnt1, GlobalData.
543                 ↪ ListaTAGs[indice].Leitora2.DateTimePicoRSSIAnt2) < 0)
544             {
545                 // Como acabou de ocorrer uma transição, seta a variável ReportON para falso para que

```

```

543 // essa tag nao receba reports por alguns segundos, evitando erro nas analises futuras.
544 GlobalData.ListaTAGs[indice].ReportON = false;
545 bool passagem = true;
546
547 Console.WriteLine("\n\nLeitora endereço: {0}", sender.Address);
548
549 // Verifica o EPC das tags que estao registradas para verificar quem entrou no ambiente,
550 // e ai passa o nome da pessoa para a funcao MonitoramentoSALA.
551 TransicaoTags.transicao_sala(sender.Address, GlobalData.ListaTAGs[indice].EPC, passagem, GlobalData.
    ↳ ListaTAGs[indice].Nome, GlobalData.ListaTAGs[indice]);
552
553 // Seta a flag que indica que esta ocorrendo uma transicao entre ambientes.
554 //LeitoraDois.FlagTrocaAmbiente = 1;
555 //GlobalData.ListaTAGs[indice].Leitora2.FlagTrocaAmbiente = 1;
556 MonitoramentoTAG.ResetaTAG(2, indice);
557
558 // Inicializa a Thread que faz o descarte de reports das tags por 2 segundos.
559 // Isso eh feito para que os dados dos reports durante a transicao da pessoa
560 // nao atrapalhem a analise.
561 Thread DescartaReportThread = new Thread() => MonitoramentoTAG.DescartaReport(indice));
562 DescartaReportThread.Start();
563 }
564
565 // Checa os tempos de cada pico para saber se a TAG entrou ou saiu do ambiente.
566 // Se a comparacao for positiva, significa que passaram primeiro pela antena 2 e depois
567 // pela antena 1, ou seja, a pessoa saiu do ambiente.
568 if (DateTime.Compare(GlobalData.ListaTAGs[indice].Leitora2.DateTimePicoRSSIAnt1, GlobalData.
    ↳ ListaTAGs[indice].Leitora2.DateTimePicoRSSIAnt2) > 0)
569 {
570 // Como acabou de ocorrer uma transicao, seta a variavel ReportON para falso para que
571 // essa tag nao receba reports por alguns segundos, evitando erro nas analises futuras.
572 GlobalData.ListaTAGs[indice].ReportON = false;
573 bool passagem = false;
574
575 Console.WriteLine("\n\nLeitora endereço: {0}", sender.Address);
576
577 // Verifica o EPC das tags que estao registradas para verificar quem saiu do ambiente,
578 // e ai passa o nome da pessoa para a funcao MonitoramentoSALA.
579 TransicaoTags.transicao_sala(sender.Address, GlobalData.ListaTAGs[indice].EPC, passagem, GlobalData.
    ↳ ListaTAGs[indice].Nome, GlobalData.ListaTAGs[indice]);
580
581 // Seta a flag que indica que esta ocorrendo uma transicao entre ambientes.
582 //LeitoraDois.FlagTrocaAmbiente = 1;
583 //GlobalData.ListaTAGs[indice].Leitora2.FlagTrocaAmbiente = 1;
584 MonitoramentoTAG.ResetaTAG(2, indice);
585
586 // Inicializa a Thread que faz o descarte de reports das tags por 2 segundos.
587 // Isso eh feito para que os dados dos reports durante a transicao da pessoa
588 // nao atrapalhem a analise.
589 Thread DescartaReportThread = new Thread() => MonitoramentoTAG.DescartaReport(indice));
590 DescartaReportThread.Start();
591 }
592 }
593
594 // Armazena o DateTime do tempo da ultima vez que foi feito uma leitura valida.
595 GlobalData.LastSeen = DateTime.Now;
596
597 //Escreve nos arquivos os dados.
598 if (tag.AntennaPortNumber == 1)
599 {
600 GlobalData.filehandler.WriteToFile(tag.Epc.ToString(), sender.Name, tag.AntennaPortNumber, tag.
    ↳ RfDopplerFrequency.ToString("0.00"), tag.PeakRssiInDbm.ToString(), "", GlobalData.
    ↳ DictSalaReunioes.Count, GlobalData.DictSalaPrincipal.Count, GlobalData.DictCorredorBaias.Count
    ↳ , GlobalData.DictAmbienteExterno.Count, GlobalData.AcionamtSalaReunioes, GlobalData.
    ↳ AcionamtSalaPrincipal, GlobalData.AcionamtCorredorBaias);
601 }
602 else
603 {
604 GlobalData.filehandler.WriteToFile(tag.Epc.ToString(), sender.Name, tag.AntennaPortNumber, tag.
    ↳ RfDopplerFrequency.ToString("0.00"), "", tag.PeakRssiInDbm.ToString(), GlobalData.
    ↳ DictSalaReunioes.Count, GlobalData.DictSalaPrincipal.Count, GlobalData.DictCorredorBaias.Count
    ↳ , GlobalData.DictAmbienteExterno.Count, GlobalData.AcionamtSalaReunioes, GlobalData.
    ↳ AcionamtSalaPrincipal, GlobalData.AcionamtCorredorBaias);
605 }
606 }
607 }
608 break;
609
610 // Caso o report da TAG seja relacionado a Leitora #3, entra aqui.
611 case "SpeedwayR-10-9F-BB.local":
612 foreach (Tag tag in report)

```

```

613     {
614
615         // Variavel do tipo Tags_TG que vai armazenar a tag que foi reportada agora.
616         Tags_TG TagAtual;
617
618         // Variavel que indica se aceita o report da TAG ou nao.
619         int TAGsReportsON = 0;
620
621         // Variaveis usadas no foreach para determinar o indice da TAG na lista de tags.
622         int contador = 0;
623         int indice = 0;
624
625         // Verifica o EPC das tags que estao registradas para verificar se o report dessa TAG
626         // deve ser aceito ou nao dependendo da variavel ReportON da TAG. Caso seja true,
627         // a variavel TAGReportON eh setada e o report eh aceito.
628         foreach (var TagRegistrada in GlobalData.ListaTAGs)
629         {
630             if (TagRegistrada.EPC == tag.Epc.ToString())
631             {
632                 TagAtual = TagRegistrada;
633                 indice = contador;
634                 if (TagAtual.ReportON == true)
635                 {
636                     TAGsReportsON = 1;
637                 }
638             }
639             contador++;
640         }
641
642         // Condicional com um filtro para validar leitura. Valor de RSSI tem que estar acima
643         // do valor do filtro e TAGsReportsON tem que ser verdadeiro.
644         if (tag.PeakRssiInDbm > GlobalData.FiltroPassaBaixaRSSI && TAGsReportsON == 1)
645         {
646             // Checa se a antena 1 que está lendo
647             if (tag.AntennaPortNumber == 1)
648             {
649                 // Se o RSSI lido do report for maior que o pico atual
650                 // de RSSI, atualiza o valor máximo de RSSI e seu tempo.
651                 if (tag.PeakRssiInDbm > GlobalData.ListaTAGs[indice].Leitora3.PicoRSSIAnt1 && GlobalData.ListaTAGs[
652                     ↪ indice].Leitora3.LockPicoRSSIAnt1 == 0)
653                 {
654                     GlobalData.ListaTAGs[indice].Leitora3.PicoRSSIAnt1 = tag.PeakRssiInDbm;
655                     GlobalData.ListaTAGs[indice].Leitora3.TempoPicoRSSIAnt1 = DateTime.Now.ToString("HH-mm-ss-fff"
656                     ↪ );
657                     GlobalData.ListaTAGs[indice].Leitora3.DatePicoRSSIAnt1 = DateTime.Now;
658
659                     // Verifica se o pico de RSSI da antena 1 é maior ou igual que o indicador RSSI de passagem.
660                     // Se for, atualiza a flag que indica que houve passagem pela primeira antena e
661                     // da "lock" nesse valor, para que nao pegue outras medidas desnecessariamente.
662                     if (GlobalData.ListaTAGs[indice].Leitora3.PicoRSSIAnt1 >= GlobalData.IndicadorRSSIPassagem)
663                     {
664                         GlobalData.ListaTAGs[indice].Leitora3.FlagPassagemAnt1 = 1;
665                         GlobalData.ListaTAGs[indice].Leitora3.LockPicoRSSIAnt1 = 1;
666                         LeitoraTres.ContadorPassagemAnt1 += 1;
667
668                         // Faz o reset das variaveis da TAG caso passe 5 segundos desde um lock e não ocorra uma passagem
669                         if (GlobalData.ListaTAGs[indice].Leitora3.LockPicoRSSIAnt2 == 0)
670                         {
671                             GlobalData.ListaTAGs[indice].TimerReset = 3;
672                             Thread TimerResetTAG = new Thread(() => MonitoramentoTAG.TimerResetTAG(indice));
673                             TimerResetTAG.Start();
674                         }
675                     }
676                 }
677             }
678
679             // Checa se a antena 2 que está lendo.
680             if (tag.AntennaPortNumber == 2)
681             {
682                 // Se o RSSI lido do report for maior que o pico atual
683                 // de RSSI, atualiza o valor máximo de RSSI e seu tempo.
684                 if (tag.PeakRssiInDbm > GlobalData.ListaTAGs[indice].Leitora3.PicoRSSIAnt2 && GlobalData.ListaTAGs[
685                     ↪ indice].Leitora3.LockPicoRSSIAnt2 == 0)
686                 {
687                     GlobalData.ListaTAGs[indice].Leitora3.PicoRSSIAnt2 = tag.PeakRssiInDbm;
688                     GlobalData.ListaTAGs[indice].Leitora3.TempoPicoRSSIAnt2 = DateTime.Now.ToString("HH-mm-ss-fff"
689                     ↪ );
690                     GlobalData.ListaTAGs[indice].Leitora3.DatePicoRSSIAnt2 = DateTime.Now;
691
692                     // Verifica se o pico de RSSI da antena 1 é maior ou igual que o indicador RSSI de passagem.

```

```

690 // Se for, atualiza a flag que indica que houve passagem pela primeira antena e
691 // da "lock" nesse valor, para que nao pegue outras medidas desnecessariamente.
692 if (GlobalData.ListaTAGs[indice].Leitora3.PicoRSSIAnt2 >= GlobalData.IndicadorRSSIPassagem)
693 {
694     GlobalData.ListaTAGs[indice].Leitora3.FlagPassagemAnt2 = 1;
695     GlobalData.ListaTAGs[indice].Leitora3.LockPicoRSSIAnt2 = 1;
696     LeitoraTres.ContadorPassagemAnt2 += 1;
697
698     // Faz o reset das variaveis da TAG caso passe 5 segundos desde um lock e não ocorra uma passagem
699     if (GlobalData.ListaTAGs[indice].Leitora3.LockPicoRSSIAnt1 == 0)
700     {
701         GlobalData.ListaTAGs[indice].TimerReset = 3;
702         Thread TimerResetTAG = new Thread(() => MonitoramentoTAG.TimerResetTAG(indice));
703         TimerResetTAG.Start();
704     }
705 }
706 }
707 }
708
709 // Checa se a flag de passagem das duas antenas foram setadas.
710 // Se sim, significa que houve a passagem da TAG pelas duas antenas.
711 // Ou seja, significa que alguem entrou ou saiu do ambiente.
712 if (GlobalData.ListaTAGs[indice].Leitora3.FlagPassagemAnt1 == 1 && GlobalData.ListaTAGs[indice].Leitora3.
    ↪ FlagPassagemAnt2 == 1)
713 {
714     // Checa os tempos de cada pico para saber se a TAG entrou ou saiu do ambiente.
715     // Se a comparacao for negativa, significa que passaram primeiro pela antena 1 e depois
716     // pela antena 2, ou seja, a pessoa entrou no ambiente.
717     if (DateTime.Compare(GlobalData.ListaTAGs[indice].Leitora3.DateTimePicoRSSIAnt1, GlobalData.
        ↪ ListaTAGs[indice].Leitora3.DateTimePicoRSSIAnt2) < 0)
718     {
719         // Como acabou de ocorrer uma transicao, seta a variavel ReportON para falso para que
720         // essa tag nao receba reports por alguns segundos, evitando erro nas analises futuras.
721         GlobalData.ListaTAGs[indice].ReportON = false;
722         bool passagem = true;
723
724         Console.WriteLine("\n\nLeitora endereço: {0}", sender.Address);
725
726         // Verifica o EPC das tags que estao registradas para verificar quem entrou no ambiente,
727         // e ai passa o nome da pessoa para a funcao MonitoramentoSALA.
728         TransicaoTags.transicao_sala(sender.Address, GlobalData.ListaTAGs[indice].EPC, passagem, GlobalData.
            ↪ ListaTAGs[indice].Nome, GlobalData.ListaTAGs[indice]);
729
730         // Seta a flag que indica que esta ocorrendo uma transicao entre ambientes.
731         //LeitoraTres.FlagTrocaAmbiente = 1;
732         //GlobalData.ListaTAGs[indice].Leitora3.FlagTrocaAmbiente = 1;
733         MonitoramentoTAG.ResetaTAG(3, indice);
734
735         // Inicializa a Thread que faz o descarte de reports das tags por 2 segundos.
736         // Isso eh feito para que os dados dos reports durante a transicao da pessoa
737         // nao atrapalhem a analise.
738         Thread DescartaReportThread = new Thread(() => MonitoramentoTAG.DescartaReport(indice));
739         DescartaReportThread.Start();
740     }
741 }
742
743 // Checa os tempos de cada pico para saber se a TAG entrou ou saiu do ambiente.
744 // Se a comparacao for positiva, significa que passaram primeiro pela antena 2 e depois
745 // pela antena 1, ou seja, a pessoa saiu do ambiente.
746 if (DateTime.Compare(GlobalData.ListaTAGs[indice].Leitora3.DateTimePicoRSSIAnt1, GlobalData.
    ↪ ListaTAGs[indice].Leitora3.DateTimePicoRSSIAnt2) > 0)
747 {
748     // Como acabou de ocorrer uma transicao, seta a variavel ReportON para falso para que
749     // essa tag nao receba reports por alguns segundos, evitando erro nas analises futuras.
750     GlobalData.ListaTAGs[indice].ReportON = false;
751     bool passagem = false;
752
753     Console.WriteLine("\n\nLeitora endereço: {0}", sender.Address);
754
755     // Verifica o EPC das tags que estao registradas para verificar quem saiu do ambiente,
756     // e ai passa o nome da pessoa para a funcao MonitoramentoSALA.
757     TransicaoTags.transicao_sala(sender.Address, GlobalData.ListaTAGs[indice].EPC, passagem, GlobalData.
        ↪ ListaTAGs[indice].Nome, GlobalData.ListaTAGs[indice]);
758
759     // Seta a flag que indica que esta ocorrendo uma transicao entre ambientes.
760     //LeitoraTres.FlagTrocaAmbiente = 1;
761     //GlobalData.ListaTAGs[indice].Leitora3.FlagTrocaAmbiente = 1;
762     MonitoramentoTAG.ResetaTAG(3, indice);
763
764     // Inicializa a Thread que faz o descarte de reports das tags por 2 segundos.
765     // Isso eh feito para que os dados dos reports durante a transicao da pessoa
766     // nao atrapalhem a analise.

```

```

766         Thread DescartaReportThread = new Thread(() => MonitoramentoTAG.DescartaReport(indice));
767         DescartaReportThread.Start();
768     }
769 }
770
771 // Armazena o DateTime do tempo da ultima vez que foi feito uma leitura valida.
772 GlobalData.LastSeen = DateTime.Now;
773
774 //Escreve nos arquivos os dados.
775 if (tag.AntennaPortNumber == 1)
776 {
777     GlobalData.filehandler.WriteToFile(tag.Epc.ToString(), sender.Name, tag.AntennaPortNumber, tag.
        ↳ RfDopplerFrequency.ToString("0.00"), tag.PeakRssiInDbm.ToString(), "", GlobalData.
        ↳ DictSalaReunioes.Count, GlobalData.DictSalaPrincipal.Count, GlobalData.DictCorredorBaias.Count
        ↳ , GlobalData.DictAmbienteExterno.Count, GlobalData.AcionamtSalaReunioes, GlobalData.
        ↳ AcionamtSalaPrincipal, GlobalData.AcionamtCorredorBaias);
778 }
779 else
780 {
781     GlobalData.filehandler.WriteToFile(tag.Epc.ToString(), sender.Name, tag.AntennaPortNumber, tag.
        ↳ RfDopplerFrequency.ToString("0.00"), "", tag.PeakRssiInDbm.ToString(), GlobalData.
        ↳ DictSalaReunioes.Count, GlobalData.DictSalaPrincipal.Count, GlobalData.DictCorredorBaias.Count
        ↳ , GlobalData.DictAmbienteExterno.Count, GlobalData.AcionamtSalaReunioes, GlobalData.
        ↳ AcionamtSalaPrincipal, GlobalData.AcionamtCorredorBaias);
782 }
783 }
784 }
785 break;
786
787 default:
788     Console.WriteLine("Caso não identificado");
789     break;
790 }
791 }
792 }
793
794
795
796 // Classe que faz o monitoramento das TAGs, verificando se a flag que indica que uma transicao entre ambientes foi setada.
797 public class MonitoramentoTAG
798 {
799
800     // Cria um timer para resetar as variaveis da TAG caso nao ocorra passagem em 5 segs.
801     public static void TimerResetTAG(int indice)
802     {
803         Thread.Sleep(2500);
804         if (GlobalData.ListaTAGs[indice].TimerReset == 1)
805         {
806             MonitoramentoTAG.ResetaTAG(1, indice);
807             Console.WriteLine("\nResetou tag leitora 1: {0}\n", GlobalData.ListaTAGs[indice].Nome);
808         }
809
810         if (GlobalData.ListaTAGs[indice].TimerReset == 2)
811         {
812             MonitoramentoTAG.ResetaTAG(2, indice);
813             Console.WriteLine("\nResetou tag leitora 2: {0}\n", GlobalData.ListaTAGs[indice].Nome);
814         }
815
816         if (GlobalData.ListaTAGs[indice].TimerReset == 3)
817         {
818             MonitoramentoTAG.ResetaTAG(3, indice);
819             Console.WriteLine("\nResetou tag leitora 3: {0}\n", GlobalData.ListaTAGs[indice].Nome);
820         }
821     }
822 }
823
824
825
826 // Reseta as variaveis da TAG.
827 public static void ResetaTAG(int var, int indice)
828 {
829     switch (var)
830     {
831     case 1:
832         // Reseta as variaveis relacionadas aos dados da primeira antena da LEITORA UM.
833         LeitoraUm.PicoRSSIAnt1 = -63;
834         LeitoraUm.PicoNegFreqDopAnt1 = 0;
835         LeitoraUm.FlagPassagemAnt1 = 0;
836         LeitoraUm.LockPicoRSSIAnt1 = 0;
837
838         // Reseta as variaveis relacionadas aos dados da segunda antena da LEITORA UM.

```

```

839     LeitoraUm.PicoRSSIAnt2 = -63;
840     LeitoraUm.PicoNegFreqDopAnt2 = 0;
841     LeitoraUm.FlagPassagemAnt2 = 0;
842     LeitoraUm.LockPicoRSSIAnt2 = 0;
843
844     // Reseta a variavel que indica a troca de ambiente da LEITORA UM.
845     LeitoraUm.FlagTrocaAmbiente = 0;
846
847     // Reseta a variavel do timer.
848     GlobalData.ListaTAGs[indice].TimerReset = 0;
849
850     // Reseta os valores da TAG.
851     GlobalData.ListaTAGs[indice].Leitora1.PicoRSSIAnt1 = -63;
852     GlobalData.ListaTAGs[indice].Leitora1.FlagPassagemAnt1 = 0;
853     GlobalData.ListaTAGs[indice].Leitora1.LockPicoRSSIAnt1 = 0;
854     GlobalData.ListaTAGs[indice].Leitora1.PicoRSSIAnt2 = -63;
855     GlobalData.ListaTAGs[indice].Leitora1.FlagPassagemAnt2 = 0;
856     GlobalData.ListaTAGs[indice].Leitora1.LockPicoRSSIAnt2 = 0;
857
858     break;
859
860
861     case 2:
862         // Reseta as variaveis relacionadas aos dados da primeira antena da LEITORA DOIS.
863         LeitoraDois.PicoRSSIAnt1 = -63;
864         LeitoraDois.PicoNegFreqDopAnt1 = 0;
865         LeitoraDois.FlagPassagemAnt1 = 0;
866         LeitoraDois.LockPicoRSSIAnt1 = 0;
867
868         // Reseta as variaveis relacionadas aos dados da segunda antena da LEITORA DOIS.
869         LeitoraDois.PicoRSSIAnt2 = -63;
870         LeitoraDois.PicoNegFreqDopAnt2 = 0;
871         LeitoraDois.FlagPassagemAnt2 = 0;
872         LeitoraDois.LockPicoRSSIAnt2 = 0;
873
874         // Reseta a variavel que indica a troca de ambiente da LEITORA DOIS.
875         LeitoraDois.FlagTrocaAmbiente = 0;
876
877         // Reseta a variavel do timer.
878         GlobalData.ListaTAGs[indice].TimerReset = 0;
879
880         // Reseta os valores da TAG.
881         GlobalData.ListaTAGs[indice].Leitora2.PicoRSSIAnt1 = -63;
882         GlobalData.ListaTAGs[indice].Leitora2.FlagPassagemAnt1 = 0;
883         GlobalData.ListaTAGs[indice].Leitora2.LockPicoRSSIAnt1 = 0;
884         GlobalData.ListaTAGs[indice].Leitora2.PicoRSSIAnt2 = -63;
885         GlobalData.ListaTAGs[indice].Leitora2.FlagPassagemAnt2 = 0;
886         GlobalData.ListaTAGs[indice].Leitora2.LockPicoRSSIAnt2 = 0;
887
888         break;
889
890
891     case 3:
892         // Reseta as variaveis relacionadas aos dados da primeira antena da LEITORA TRES.
893         LeitoraTres.PicoRSSIAnt1 = -63;
894         LeitoraTres.PicoNegFreqDopAnt1 = 0;
895         LeitoraTres.FlagPassagemAnt1 = 0;
896         LeitoraTres.LockPicoRSSIAnt1 = 0;
897
898         // Reseta as variaveis relacionadas aos dados da segunda antena da LEITORA TRES.
899         LeitoraTres.PicoRSSIAnt2 = -63;
900         LeitoraTres.PicoNegFreqDopAnt2 = 0;
901         LeitoraTres.FlagPassagemAnt2 = 0;
902         LeitoraTres.LockPicoRSSIAnt2 = 0;
903
904         // Reseta a variavel que indica a troca de ambiente da LEITORA TRES.
905         LeitoraTres.FlagTrocaAmbiente = 0;
906
907         // Reseta a variavel do timer.
908         GlobalData.ListaTAGs[indice].TimerReset = 0;
909
910         // Reseta os valores da TAG.
911         GlobalData.ListaTAGs[indice].Leitora3.PicoRSSIAnt1 = -63;
912         GlobalData.ListaTAGs[indice].Leitora3.FlagPassagemAnt1 = 0;
913         GlobalData.ListaTAGs[indice].Leitora3.LockPicoRSSIAnt1 = 0;
914         GlobalData.ListaTAGs[indice].Leitora3.PicoRSSIAnt2 = -63;
915         GlobalData.ListaTAGs[indice].Leitora3.FlagPassagemAnt2 = 0;
916         GlobalData.ListaTAGs[indice].Leitora3.LockPicoRSSIAnt2 = 0;
917
918         break;
919

```

```

920         default:
921             Console.WriteLine("Caso não identificado");
922             break;
923     }
924 }
925
926
927
928 // Funcao que faz o descarte dos reports durante 2 segundos.
929 // Seta TAGsReportsON para zero e faz com que AnalisaDadosTAGs
930 // descarte os reports durante 2 segundos, depois seta para 1 de novo.
931 public static void DescartaReport(int indice)
932 {
933     Thread.Sleep(3000);
934     GlobalData.ListaTAGs[indice].ReportON = true;
935 }
936 }
937
938
939 // Classe que lida com o armazenamento dos dados em arquivos.
940 public class FileHandler
941 {
942     protected string filePathCsv;
943     protected string filePathTxt;
944
945     public void SetFileHandler()
946     {
947         filePathCsv = @Environment.GetFolderPath(Environment.SpecialFolder.Desktop) + "\\logcsv_" + DateTime.Now.ToString("
948             ↪ yyyyMMdd_HH-mm-ss") + ".csv";
949
950         //filePathTxt = @Environment.GetFolderPath(Environment.SpecialFolder.Desktop) + "\\logtxt_" + DateTime.Now.ToString("
951             ↪ yyyyMMdd_HH-mm-ss") + ".txt";
952     }
953
954
955     public void WriteToFile(string epc, string reader, ushort ant, string doppler_frequency, string peak_rssi1, string peak_rssi2, int
956         ↪ contadorSalaPrincipal, int contadorSalaReunioes, int contadorCorredorPrincipal, int contadorAreaExterna, int ReleReunioes,
957         ↪ int RelePrincipal, int ReleBaia)
958     {
959         // Set File parameters
960         //string csvSeparator = "\t";
961         string csvSeparator = ",";
962
963         StringBuilder streamOutput = new StringBuilder();
964
965         //var antenna = Tuple.Create<string, ushort>(reader, ant);
966         //doppler_frequency = String.Concat(csvsignal, doppler_frequency, csvsignal);
967
968         string[][] dataOutput = new string[][]{
969             new string[] { epc, reader, ant.ToString(), DateTime.Now.ToString("dd/MM/yyyy; HH:mm:ss:fff"),
970                 ↪ doppler_frequency, peak_rssi1, peak_rssi2, GlobalData.IndicadorRSSIPassagem.ToString(),
971                 ↪ contadorSalaPrincipal.ToString(), contadorSalaReunioes.ToString(), contadorCorredorPrincipal.
972                 ↪ ToString(), contadorAreaExterna.ToString(), ReleReunioes.ToString(), RelePrincipal.ToString(),
973                 ↪ ReleBaia.ToString() }
974 };
975
976         int length = dataOutput.GetLength(0);
977         for (int i = 0; i < length; i++)
978             streamOutput.AppendLine(string.Join(csvSeparator, dataOutput[i]));
979
980         //string TxtData = string.Format("Pico RSSI da ANTENA 1 : {0} \nTempo do pico da ANTENA 1 : {1} \nPico negativo da
981             ↪ Frequencia Doppler da ANTENA 1 : {2} \nTempo do pico negativo da Frequencia Doppler da ANTENA 1 : {3} \n\
982             ↪ nPico RSSI da ANTENA 2 : {4} \nTempo do pico da ANTENA 2 : {5} \nPico negativo da Frequencia Doppler da
983             ↪ ANTENA 2 : {6} \nTempo do pico negativo da Frequencia Doppler da ANTENA 2 : {7}", GlobalData.PicoRSSIAnt1,
984             ↪ GlobalData.TempoPicoRSSIAnt1, GlobalData.PicoNegFreqDopAnt1, GlobalData.TempoPicoNegFreqDopAnt1,
985             ↪ GlobalData.PicoRSSIAnt2, GlobalData.TempoPicoRSSIAnt2, GlobalData.PicoNegFreqDopAnt2, GlobalData.
986             ↪ TempoPicoNegFreqDopAnt2);
987
988         // Appends more lines to the csv file.
989         // Faz um loop que verifica se estao escrevendo no arquivo antes de tentar escrever para nao dar conflito.
990         bool loop = true;
991         while (loop)
992         {
993             if (!GlobalData.EscrevendoArquivo)
994             {
995                 GlobalData.EscrevendoArquivo = true;
996                 File.AppendAllText(filePathCsv, streamOutput.ToString());
997                 loop = false;
998                 GlobalData.EscrevendoArquivo = false;
999             }
1000         }

```

```

987     }
988     }
989     //File.WriteAllText(filePathTxt, TxtData);
990 }
991
992
993
994 public void CreateFile()
995 {
996     // Set File parameters
997     //string csvSeparator = "\t";
998     string csvSeparator = ",";
999     StringBuilder streamOutput = new StringBuilder();
1000
1001     string[][] dataOutput = new string[][]{
1002         new string[]{"EPC", "Leitora", "Antena", "Data", "Horario","Frequencia Doppler", "RSSI Ant1", "RSSI
           ↳ Ant2", "IndicadorPassagem", "Sala Reunioes", "Sala Principal", "Corredor de Baias", "Area
           ↳ Externa", "Rele S. Reunioes", "Rele S. Principal", "Rele C. Baias" }
1003     };
1004     int length = dataOutput.GetLength(0);
1005     for (int i = 0; i < length; i++)
1006         streamOutput.AppendLine(string.Join(csvSeparator, dataOutput[i]));
1007
1008     //string TxtData = string.Format("Pico RSSI da ANTENA 1 : {0} \nTempo do pico da ANTENA 1 : {1} \nPico negativo da
           ↳ Frequencia Doppler da ANTENA 1 : {2} \nTempo do pico negativo da Frequencia Doppler da ANTENA 1 : {3} \n\
           ↳ nPico RSSI da ANTENA 2 : {4} \nTempo do pico da ANTENA 2 : {5} \nPico negativo da Frequencia Doppler da
           ↳ ANTENA 2 : {6} \nTempo do pico negativo da Frequencia Doppler da ANTENA 2 : {7}", GlobalData.PicoRSSIAnt1,
           ↳ GlobalData.TempoPicoRSSIAnt1, GlobalData.PicoNegFreqDopAnt1, GlobalData.TempoPicoNegFreqDopAnt1,
           ↳ GlobalData.PicoRSSIAnt2, GlobalData.TempoPicoRSSIAnt2, GlobalData.PicoNegFreqDopAnt2, GlobalData.
           ↳ TempoPicoNegFreqDopAnt2);
1009
1010     // Create and write the csv and txt file.
1011     File.WriteAllText(filePathCsv, streamOutput.ToString());
1012     //File.WriteAllText(filePathTxt, TxtData);
1013 }
1014 }
1015 }

```

VariaveisProgram.cs

```
1 ///////////////////////////////////////////////////////////////////
2 //
3 // Trabalho de Graduação 2 – Leonardo Pereira e Kenneth Lui
4 // Universidade de Brasília – UnB
5 // Novembro de 2020
6 //
7 // Variaveis relacionadas ao Reader, Antena 1, Antena 2 e flags do programa.
8 //
9 ///////////////////////////////////////////////////////////////////
10
11
12 using System;
13 using Impinj.OctaneSdk;
14 using System.Collections;
15 using System.Collections.Generic;
16 using System.Text;
17 using System.IO;
18 using System.Threading;
19 using System.Security.Cryptography.X509Certificates;
20
21
22
23 namespace TeGe2
24 {
25
26     class GlobalData
27     {
28         // Cria um objeto filehandler
29         public static FileHandler filehandler = new FileHandler();
30
31         //Variavel que guarda o valor minimo de RSSI para considerar a leitura.
32         public static int FiltroPassaBaixaRSSI = -58;
33
34         // Variavel que armazena o valor de RSSI que indica que alguém passou na frente da antena.
35         public static int IndicadorRSSIPassagem = -47;
36
37         // Variavel que armazena o DateTime da ultima vez que a TAG foi lida
38         public static DateTime LastSeen = DateTime.Now;
39
40         // Flags relacionadas ao acionamento do atuador nos ambientes
41         public static int AcionamtSalaPrincipal = 0;
42         public static int AcionamtSalaReunioes = 0;
43         public static int AcionamtCorredorBaias = 0;
44
45         // Variaveis que armazenam o peso que indica se o rele sera acionado ou nao
46         public static double contaPessoaAmbPrincipal = 0;
47         public static double contaPessoaAmbReunioes = 0;
48         public static double contaPessoaAmbBaias = 0;
49
50         // Flag que indica se o programa continua rodando ou não.
51         // Éutilizado para fechar a thread MonitoraTag quando algumem pressiona Enter
52         public static int FlagPrograma;
53
54         // Flag que indica se estao escrevendo no arquivo
55         public static bool EscrevendoArquivo = false;
56
57         //Dicionário com todas as EPCs das tags e as respectivas salas em que se encontram
58         public static Dictionary<string, int> tag_sala = new Dictionary<string, int>();
59
60         // Cria uma lista de objetos Tags_TG com os EPCs e nomes das pessoas.
61         // Adicione as novas TAGs colocando seu EPC e o nome relacionado.
62         public static List<Tags_TG> ListaTAGs = new List<Tags_TG>
63         {
64             new Tags_TG{EPC = "E200 001B 2609 0146 2580 7745", Nome = "Leonardo" , Genero = "M", Altura = 1.80, Peso = 70 ,
65                 ↳ Ambiente = 0, ReportON = true, TimerReset = 0},
66             new Tags_TG{EPC = "E200 001B 2609 0146 2700 7715", Nome = "Fernanda" , Genero = "F", Altura = 1.70, Peso = 55 ,
67                 ↳ Ambiente = 0, ReportON = true, TimerReset = 0},
68             new Tags_TG{EPC = "E200 001B 2609 0146 2770 76FD", Nome = "Kenneth" , Genero = "M", Altura = 1.75, Peso = 75 ,
69                 ↳ Ambiente = 0, ReportON = true, TimerReset = 0},
70             new Tags_TG{EPC = "E200 001B 2609 0146 2630 7739", Nome = "Jhon" , Genero = "M", Altura = 1.60, Peso = 63 , Ambiente =
71                 ↳ 0, ReportON = true, TimerReset = 0},
72             new Tags_TG{EPC = "E200 001B 2609 0146 2780 76F5", Nome = "Marcelo" , Genero = "M", Altura = 1.80, Peso = 70 ,
73                 ↳ Ambiente = 0, ReportON = true, TimerReset = 0},
74             new Tags_TG{EPC = "E200 001B 2609 0146 2710 7719", Nome = "Gabriela" , Genero = "F", Altura = 1.80, Peso = 70 ,
75                 ↳ Ambiente = 0, ReportON = true, TimerReset = 0},
76             new Tags_TG{EPC = "E200 001B 2609 0146 2650 772D", Nome = "Julia" , Genero = "F", Altura = 1.80, Peso = 70 , Ambiente =
77                 ↳ 0, ReportON = true, TimerReset = 0},
78             new Tags_TG{EPC = "E200 001B 2609 0145 2880 76A4", Nome = "Guilherme" , Genero = "M", Altura = 1.80, Peso = 70 ,
79                 ↳ Ambiente = 0, ReportON = true, TimerReset = 0}
80         }
81     }
82 }
```

```

72     };
73
74     // Cria dicionarios dos ambientes para fazer a contagem de pessoas por ambiente e inicializa eles.
75     // Nesse caso, todas TAGs estao sendo inicializadas no ambiente externo.
76     public static Dictionary<string, Tags_TG> DictAmbienteExterno = new Dictionary<string, Tags_TG>(){
77         { "E200 001B 2609 0146 2580 7745", ListaTAGs[0] },
78         { "E200 001B 2609 0146 2700 7715", ListaTAGs[1] },
79         { "E200 001B 2609 0146 2770 76FD", ListaTAGs[2] },
80         { "E200 001B 2609 0146 2630 7739", ListaTAGs[3] },
81         { "E200 001B 2609 0146 2780 76F5", ListaTAGs[4] },
82         { "E200 001B 2609 0146 2710 7719", ListaTAGs[5] },
83         { "E200 001B 2609 0146 2650 772D", ListaTAGs[6] },
84         { "E200 001B 2609 0145 2880 76A4", ListaTAGs[7] }
85     };
86     public static Dictionary<string, Tags_TG> DictSalaReunioes = new Dictionary<string, Tags_TG>();
87     public static Dictionary<string, Tags_TG> DictCorredorBaiais = new Dictionary<string, Tags_TG>();
88     public static Dictionary<string, Tags_TG> DictSalaPrincipal = new Dictionary<string, Tags_TG>();
89
90     public static int Contador_SalaReunioes = DictSalaReunioes.Count;
91     public static int Contador_SalaPrincipal = DictSalaPrincipal.Count;
92     public static int Contador_CorredorBaiais = DictCorredorBaiais.Count;
93
94 }
95
96
97
98 class LeitoraUm
99 {
100     // Flag que armazena se houve uma transição entre ambientes (se saiu ou entrou)
101     public static int FlagTrocaAmbiente = 0;
102
103     //Variaveis relacionadas aos dados da primeira antena.
104     public static double PicoRSSIAnt1 = -63;
105     public static string TempoPicoRSSIAnt1;
106     public static DateTime DateTimePicoRSSIAnt1;
107     public static double PicoNegFreqDopAnt1 = 0;
108     public static string TempoPicoNegFreqDopAnt1;
109     public static DateTime DateTimePicoNegFreqDopAnt1;
110     public static int FlagPassagemAnt1 = 0;
111     public static int LockPicoRSSIAnt1 = 0;
112     public static int ContadorPassagemAnt1 = 0;
113
114     //Variaveis relacionadas aos dados da segunda antena.
115     public static double PicoRSSIAnt2 = -63;
116     public static string TempoPicoRSSIAnt2;
117     public static DateTime DateTimePicoRSSIAnt2;
118     public static double PicoNegFreqDopAnt2 = 0;
119     public static string TempoPicoNegFreqDopAnt2;
120     public static DateTime DateTimePicoNegFreqDopAnt2;
121     public static int FlagPassagemAnt2 = 0;
122     public static int LockPicoRSSIAnt2 = 0;
123     public static int ContadorPassagemAnt2 = 0;
124 }
125
126
127
128 class LeitoraDois
129 {
130     // Flag que armazena se houve uma transição entre ambientes (se saiu ou entrou)
131     public static int FlagTrocaAmbiente = 0;
132
133     //Variaveis relacionadas aos dados da primeira antena.
134     public static double PicoRSSIAnt1 = -63;
135     public static string TempoPicoRSSIAnt1;
136     public static DateTime DateTimePicoRSSIAnt1;
137     public static double PicoNegFreqDopAnt1 = 0;
138     public static string TempoPicoNegFreqDopAnt1;
139     public static DateTime DateTimePicoNegFreqDopAnt1;
140     public static int FlagPassagemAnt1 = 0;
141     public static int LockPicoRSSIAnt1 = 0;
142     public static int ContadorPassagemAnt1 = 0;
143
144     //Variaveis relacionadas aos dados da segunda antena.
145     public static double PicoRSSIAnt2 = -63;
146     public static string TempoPicoRSSIAnt2;
147     public static DateTime DateTimePicoRSSIAnt2;
148     public static double PicoNegFreqDopAnt2 = 0;
149     public static string TempoPicoNegFreqDopAnt2;
150     public static DateTime DateTimePicoNegFreqDopAnt2;
151     public static int FlagPassagemAnt2 = 0;
152     public static int LockPicoRSSIAnt2 = 0;

```

```

153     public static int ContadorPassagemAnt2 = 0;
154 }
155
156
157
158 class LeitoraTres
159 {
160     // Flag que armazena se houve uma transição entre ambientes (se saiu ou entrou)
161     public static int FlagTrocaAmbiente = 0;
162
163     //Variaveis relacionadas aos dados da primeira antena.
164     public static double PicoRSSIAnt1 = -63;
165     public static string TempoPicoRSSIAnt1;
166     public static DateTime DateTimePicoRSSIAnt1;
167     public static double PicoNegFreqDopAnt1 = 0;
168     public static string TempoPicoNegFreqDopAnt1;
169     public static DateTime DateTimePicoNegFreqDopAnt1;
170     public static int FlagPassagemAnt1 = 0;
171     public static int LockPicoRSSIAnt1 = 0;
172     public static int ContadorPassagemAnt1 = 0;
173
174     //Variaveis relacionadas aos dados da segunda antena.
175     public static double PicoRSSIAnt2 = -63;
176     public static string TempoPicoRSSIAnt2;
177     public static DateTime DateTimePicoRSSIAnt2;
178     public static double PicoNegFreqDopAnt2 = 0;
179     public static string TempoPicoNegFreqDopAnt2;
180     public static DateTime DateTimePicoNegFreqDopAnt2;
181     public static int FlagPassagemAnt2 = 0;
182     public static int LockPicoRSSIAnt2 = 0;
183     public static int ContadorPassagemAnt2 = 0;
184 }
185
186
187
188 public static class ConstantesReader
189 {
190     public const string ReaderHostname1 = "SpeedwayR-10-9F-3F.local";
191     public const string ReaderHostname2 = "SpeedwayR-10-9F-C8.local";
192     public const string ReaderHostname3 = "SpeedwayR-10-9F-BB.local";
193 }
194
195
196
197 static class ConstantesAmbiente
198 {
199     public const int AmbienteExterno = 0;
200     public const int SalaPrincipal = 1;
201     public const int SalaDeReunioes = 2;
202     public const int CorredorDeBaiais = 3;
203 }
204
205 public class Leitoras
206 {
207     // Flag que armazena se houve uma transição entre ambientes (se saiu ou entrou)
208     public int FlagTrocaAmbiente = 0;
209
210     //Variaveis relacionadas aos dados da primeira antena.
211     public double PicoRSSIAnt1 = -63;
212     public string TempoPicoRSSIAnt1;
213     public DateTime DateTimePicoRSSIAnt1;
214     public int FlagPassagemAnt1 = 0;
215     public int LockPicoRSSIAnt1 = 0;
216
217     //Variaveis relacionadas aos dados da segunda antena.
218     public double PicoRSSIAnt2 = -63;
219     public string TempoPicoRSSIAnt2;
220     public DateTime DateTimePicoRSSIAnt2;
221     public int FlagPassagemAnt2 = 0;
222     public int LockPicoRSSIAnt2 = 0;
223 }
224
225 public class Tags_TG
226 {
227     public string EPC { get; set; }
228     public string Nome { get; set; }
229     public string Genero { get; set; }
230     public double Altura { get; set; }
231     public double Peso { get; set; }
232     public int Ambiente { get; set; }
233     public bool ReportON { get; set; }

```

```
234     public int TimerReset { get; set; }
235
236     public Leitoras Leitora1 = new Leitoras();
237     public Leitoras Leitora2 = new Leitoras();
238     public Leitoras Leitora3 = new Leitoras();
239 }
240
241 }
```

Transicao.cs

```

1  //////////////////////////////////////
2  //
3  // Trabalho de Graduação 2 – Leonardo Pereira e Kenneth Lui
4  // Universidade de Brasília – UnB
5  // Novembro de 2020
6  //
7  // Classes e funcoes responsaveis pelas transicoes das TAGs entrem ambientes
8  //
9  //////////////////////////////////////
10
11
12
13 using System;
14 using System.Collections;
15 using System.Collections.Generic;
16 using Impinj.OctaneSdk;
17 using System.Security.Cryptography.X509Certificates;
18 using System.Linq;
19 using System.Text;
20 using System.Threading.Tasks;
21 using TeGe;
22
23 namespace TeGe2
24 {
25     public class TransicaoTags
26     {
27
28         // Faz a transicao dos ambientes de uma tag, dependendo da variavel "passagem" e da leitora que leu
29         public static void transicao_sala(string leitora, string tag_epc, bool passagem, string nome, Tags_TG tag)
30         {
31             switch (leitora)
32             {
33                 case "SpeedwayR-10-9F-C8.local":
34                     if (passagem == true)
35                     {
36                         GlobalData.tag_sala[tag_epc] = ConstantesAmbiente.SalaPrincipal;
37                         TransicaoTags.transicaoRemove(tag_epc, tag.Ambiente);
38                         TransicaoTags.transicaoADD(tag_epc, 1, tag);
39                         Console.WriteLine("{0} realizou uma transicao para a Sala Principal", nome);
40                         TransicaoTags.DisplayAmbientes();
41                     }
42                     else
43                     {
44                         GlobalData.tag_sala[tag_epc] = ConstantesAmbiente.AmbienteExterno;
45                         TransicaoTags.transicaoRemove(tag_epc, tag.Ambiente);
46                         TransicaoTags.transicaoADD(tag_epc, 0, tag);
47                         Console.WriteLine("{0} realizou uma transicao para o Ambiente Externo", nome);
48                         TransicaoTags.DisplayAmbientes();
49                     }
50                     break;
51
52                 case "SpeedwayR-10-9F-3F.local":
53                     if (passagem == true)
54                     {
55                         GlobalData.tag_sala[tag_epc] = ConstantesAmbiente.SalaDeReunioes;
56                         TransicaoTags.transicaoRemove(tag_epc, tag.Ambiente);
57                         TransicaoTags.transicaoADD(tag_epc, 2, tag);
58                         Console.WriteLine("{0} realizou uma transicao para a Sala De Reuniões", nome);
59                         TransicaoTags.DisplayAmbientes();
60                     }
61                     else
62                     {
63                         GlobalData.tag_sala[tag_epc] = ConstantesAmbiente.SalaPrincipal;
64                         TransicaoTags.transicaoRemove(tag_epc, tag.Ambiente);
65                         TransicaoTags.transicaoADD(tag_epc, 1, tag);
66                         Console.WriteLine("{0} realizou uma transicao para a Sala Principal", nome);
67                         TransicaoTags.DisplayAmbientes();
68                     }
69                     break;
70
71                 case "SpeedwayR-10-9F-BB.local":
72                     if (passagem == true)
73                     {
74                         GlobalData.tag_sala[tag_epc] = ConstantesAmbiente.CorredorDeBaias;
75                         TransicaoTags.transicaoRemove(tag_epc, tag.Ambiente);
76                         TransicaoTags.transicaoADD(tag_epc, 3, tag);
77                         Console.WriteLine("{0} realizou uma transicao para o Corredor De Baias", nome);
78                         TransicaoTags.DisplayAmbientes();
79                     }

```

```

80     else
81     {
82         GlobalData.tag_sala[tag_epc] = ConstantesAmbiente.SalaPrincipal;
83         TransicaoTags.transicaoRemove(tag_epc, tag.Ambiente);
84         TransicaoTags.transicaoADD(tag_epc, 1, tag);
85         Console.WriteLine("{0} realizou uma transicao para a Sala Principal", nome);
86         TransicaoTags.DisplayAmbientes();
87     }
88     break;
89
90     default:
91         Console.Write("Caso não identificado");
92         break;
93     }
94 }
95
96
97
98 // Funcao responsavel por fazer a remocao da TAG no dicionario da sala anterior a transicao
99 public static void transicaoRemove(string EPC, int sala_anterior)
100 {
101     //double contaPessoaAmb = 0;
102     switch (sala_anterior)
103     {
104     case 0:
105
106         GlobalData.DictAmbienteExterno.Remove(EPC);
107         break;
108
109     case 1:
110
111         GlobalData.DictSalaPrincipal.Remove(EPC);
112         GlobalData.contaPessoaAmbPrincipal = 0;
113
114         foreach (var TagAmb in GlobalData.DictSalaPrincipal.Values)
115         {
116             if (TagAmb.Genero == "M")
117             {
118                 GlobalData.contaPessoaAmbPrincipal += 1;
119             }
120             if (TagAmb.Genero == "F")
121             {
122                 GlobalData.contaPessoaAmbPrincipal += 0.75;
123             }
124         }
125
126
127         if (GlobalData.contaPessoaAmbPrincipal <= 2 && GlobalData.AcionamtSalaPrincipal == 1)
128         {
129             Console.WriteLine("\n*Desacionamento do ventilador na Sala Principal*\n");
130             GlobalData.AcionamtSalaPrincipal = 0;
131             SerialCom.Controle_Ar("Sala Principal");
132         }
133         break;
134
135     case 2:
136
137         GlobalData.DictSalaReunioes.Remove(EPC);
138         GlobalData.contaPessoaAmbReunioes = 0;
139
140         foreach (var TagAmb in GlobalData.DictSalaReunioes.Values)
141         {
142             if (TagAmb.Genero == "M")
143             {
144                 GlobalData.contaPessoaAmbReunioes += 1;
145             }
146             if (TagAmb.Genero == "F")
147             {
148                 GlobalData.contaPessoaAmbReunioes += 0.75;
149             }
150         }
151
152         if (GlobalData.contaPessoaAmbReunioes <= 2 && GlobalData.AcionamtSalaReunioes == 1)
153         {
154             Console.WriteLine("\n*Desacionamento do ventilador na Sala de Reunioes*\n");
155             GlobalData.AcionamtSalaReunioes = 0;
156             SerialCom.Controle_Ar("Sala Reuniões");
157         }
158         break;
159
160     case 3:

```

```

161 GlobalData.DictCorredorBaiais.Remove(EPC);
162 GlobalData.contaPessoaAmbBaiais = 0;
163
164
165 foreach (var TagAmb in GlobalData.DictCorredorBaiais.Values)
166 {
167     if (TagAmb.Genero == "M")
168     {
169         GlobalData.contaPessoaAmbBaiais += 1;
170     }
171     if (TagAmb.Genero == "F")
172     {
173         GlobalData.contaPessoaAmbBaiais += 0.75;
174     }
175 }
176
177 if (GlobalData.contaPessoaAmbBaiais <= 2 && GlobalData.AcionamtCorredorBaiais == 1)
178 {
179     Console.WriteLine("\n*Desacionamento do ventilador no Corredor de Baiais*\n");
180     GlobalData.AcionamtCorredorBaiais = 0;
181     SerialCom.Controle_Ar("Corredor de Baiais");
182 }
183 break;
184
185 default:
186     Console.Write("Caso não identificado");
187     break;
188 }
189 }
190
191
192
193 // Funcao responsavel por adicionar a TAG no dicionario da sala para qual a TAG fez a transicao
194 public static void transicaoADD(string EPC, int sala_atual, Tags_TG tag)
195 {
196     switch (sala_atual)
197     {
198     case 0:
199
200         GlobalData.DictAmbienteExterno.Add(EPC, tag);
201         tag.Ambiente = 0;
202         break;
203
204     case 1:
205
206         GlobalData.DictSalaPrincipal.Add(EPC, tag);
207         tag.Ambiente = 1;
208         GlobalData.contaPessoaAmbPrincipal = 0;
209
210         foreach (var TagAmb in GlobalData.DictSalaPrincipal.Values)
211         {
212             if (TagAmb.Genero == "M")
213             {
214                 GlobalData.contaPessoaAmbPrincipal += 1;
215             }
216             if (TagAmb.Genero == "F")
217             {
218                 GlobalData.contaPessoaAmbPrincipal += 0.75;
219             }
220         }
221
222         if (GlobalData.contaPessoaAmbPrincipal > 2 && GlobalData.AcionamtSalaPrincipal == 0)
223         {
224             Console.WriteLine("\n*Acionamento do ventilador na Sala Principal*\n");
225             GlobalData.AcionamtSalaPrincipal = 1;
226             SerialCom.Controle_Ar("Sala Principal");
227         }
228         break;
229
230     case 2:
231
232         GlobalData.DictSalaReunioes.Add(EPC, tag);
233         tag.Ambiente = 2;
234         GlobalData.contaPessoaAmbReunioes = 0;
235
236         foreach (var TagAmb in GlobalData.DictSalaReunioes.Values)
237         {
238             if (TagAmb.Genero == "M")
239             {
240                 GlobalData.contaPessoaAmbReunioes += 1;
241             }

```

```

242     if (TagAmb.Genero == "F")
243     {
244         GlobalData.contaPessoaAmbReunioes += 0.75;
245     }
246 }
247
248 if (GlobalData.contaPessoaAmbReunioes > 2 && GlobalData.AcionamtSalaReunioes == 0)
249 {
250     Console.WriteLine("\n*Acionamento do ventilador na Sala de Reunioes*\n");
251     GlobalData.AcionamtSalaReunioes = 1;
252     SerialCom.Controle_Ar("Sala Reunioes");
253 }
254 break;
255
256 case 3:
257
258     GlobalData.DictCorredorBaiais.Add(EPC, tag);
259     tag.Ambiente = 3;
260     GlobalData.contaPessoaAmbBaiais = 0;
261
262     foreach (var TagAmb in GlobalData.DictCorredorBaiais.Values)
263     {
264         if (TagAmb.Genero == "M")
265         {
266             GlobalData.contaPessoaAmbBaiais += 1;
267         }
268         if (TagAmb.Genero == "F")
269         {
270             GlobalData.contaPessoaAmbBaiais += 0.75;
271         }
272     }
273
274     if (GlobalData.contaPessoaAmbBaiais > 2 && GlobalData.AcionamtCorredorBaiais == 0)
275     {
276         Console.WriteLine("\n*Acionamento do ventilador no Corredor de Baiais*\n");
277         GlobalData.AcionamtCorredorBaiais = 1;
278         SerialCom.Controle_Ar("Corredor de Baiais");
279     }
280     break;
281
282     default:
283         Console.WriteLine("Caso não identificado");
284         break;
285 }
286 }
287
288
289 // Funcao que mostra a quantidade de pessoas em cada ambiente, assim como seus EPCs e Nomes
290 public static void DisplayAmbientes()
291 {
292     Console.WriteLine("+++++++\n");
293
294     Console.WriteLine("- Numero de pessoas na Sala principal: {0}", GlobalData.DictSalaPrincipal.Count);
295     for (int i = 0; i < GlobalData.DictSalaPrincipal.Count; i++)
296     {
297         Console.WriteLine("\tEPC: {0}, Nome: {1}", GlobalData.DictSalaPrincipal.Keys.ElementAt(i), GlobalData.DictSalaPrincipal[
                ↪ GlobalData.DictSalaPrincipal.Keys.ElementAt(i)].Nome);
298     }
299
300     Console.WriteLine("- Numero de pessoas no Corredor de Baiais: {0}", GlobalData.DictCorredorBaiais.Count);
301     for (int i = 0; i < GlobalData.DictCorredorBaiais.Count; i++)
302     {
303         Console.WriteLine("\tEPC: {0}, Nome: {1}", GlobalData.DictCorredorBaiais.Keys.ElementAt(i), GlobalData.DictCorredorBaiais[
                ↪ GlobalData.DictCorredorBaiais.Keys.ElementAt(i)].Nome);
304     }
305
306     Console.WriteLine("- Numero de pessoas na Sala de reunioes: {0}", GlobalData.DictSalaReunioes.Count);
307     for (int i = 0; i < GlobalData.DictSalaReunioes.Count; i++)
308     {
309         Console.WriteLine("\tEPC: {0}, Nome: {1}", GlobalData.DictSalaReunioes.Keys.ElementAt(i), GlobalData.DictSalaReunioes[
                ↪ GlobalData.DictSalaReunioes.Keys.ElementAt(i)].Nome);
310     }
311
312     Console.WriteLine("- Numero de pessoas no Ambiente Externo: {0}", GlobalData.DictAmbienteExterno.Count);
313     for (int i = 0; i < GlobalData.DictAmbienteExterno.Count; i++)
314     {
315         Console.WriteLine("\tEPC: {0}, Nome: {1}", GlobalData.DictAmbienteExterno.Keys.ElementAt(i), GlobalData.DictAmbienteExterno[
                ↪ GlobalData.DictAmbienteExterno.Keys.ElementAt(i)].Nome);
316     }
317     Console.WriteLine("\n+++++++\n");
318 }

```

319 }
320 }



SerialCom.cs

```

1  ////////////////////////////////////////////////////////////////////
2  //
3  // Trabalho de Graduação 2 – Leonardo Pereira e Kenneth Lui
4  // Universidade de Brasília – UnB
5  // Novembro de 2020
6  //
7  // Integracao com o arduino
8  //
9  ////////////////////////////////////////////////////////////////////
10
11
12
13 using System;
14 using System.Collections.Generic;
15 using System.IO;
16 using System.Linq;
17 using System.Threading;
18 using System.IO.Ports;
19 using TeGe2;
20 using System.Security.Cryptography;
21
22 namespace TeGe
23 {
24     class SerialCom
25     {
26         static int estado;
27         static SerialPort _serialPort;
28         public static void Controle_Ar(string sala)
29         {
30             string comando;
31
32             _serialPort = new SerialPort();
33             _serialPort.PortName = "COM3";
34             _serialPort.BaudRate = 9600;
35             _serialPort.Open();
36             //_serialPort.WriteLine("0");
37
38             //int Contador_SalaReunioes = GlobalData.DictSalaReunioes.Count;
39             //int Contador_SalaPrincipal = GlobalData.DictSalaPrincipal.Count;
40             //int Contador_CorredorBaiais = GlobalData.DictCorredorBaiais.Count;
41
42             estado = 1;
43
44             //while (GlobalData.FlagPrograma == 1)
45             //{
46             //Console.WriteLine("Ligar: <0> \t Desligar: <1> \t Sair: <quit>");
47             //comando = Console.ReadLine();
48
49             //Contador_SalaReunioes = GlobalData.DictSalaReunioes.Count;
50             //Contador_SalaPrincipal = GlobalData.DictSalaPrincipal.Count;
51             //Contador_CorredorBaiais = GlobalData.DictCorredorBaiais.Count;
52
53             switch (sala)
54             {
55                 case "Sala Principal":
56                     //Console.WriteLine("Ligar LED");
57                     estado = GlobalData.AcionamtSalaPrincipal;
58                     comando = EscreveSerial("P", estado.ToString());
59                     _serialPort.Write(comando);
60                     break;
61
62                 case "Sala Reuniões":
63                     //Console.WriteLine("Ligar LED");
64                     estado = GlobalData.AcionamtSalaReunioes;
65                     comando = EscreveSerial("R", estado.ToString());
66                     _serialPort.Write(comando);
67                     break;
68
69                 case "Corredor de Baias":
70                     estado = GlobalData.AcionamtCorredorBaiais;
71                     comando = EscreveSerial("C", estado.ToString());
72                     _serialPort.Write(comando);
73                     break;
74
75                 default:
76                     //Console.WriteLine("Comando não identificado");
77                     break;
78             }
79             //}

```

```
80     _serialPort.Close();
81 }
82
83 public static string EscreveSerial(string sala, string estado)
84 {
85     String cmd = "<" + sala + ";" + estado + ">";
86     return cmd;
87 }
88 }
89 }
```

Program.cs

```
1 ///////////////////////////////////////////////////////////////////
2 //
3 // Trabalho de Graduação 2 – Leonardo Pereira e Kenneth Lui
4 // Universidade de Brasília – UnB
5 // Novembro de 2020
6 //
7 // Interface grafica
8 //
9 ///////////////////////////////////////////////////////////////////
10
11
12 using System;
13 using System.Collections.Generic;
14 using System.Linq;
15 using System.Threading.Tasks;
16 using System.Windows.Forms;
17
18 namespace TeGe2
19 {
20     public static class ProgramGUI
21     {
22         /// <summary>
23         /// Ponto de entrada principal para o aplicativo.
24         /// </summary>
25         [STAThread]
26         public static void MainGUI()
27         {
28             Application.EnableVisualStyles();
29             Application.SetCompatibleTextRenderingDefault(false);
30             Application.Run(new Form1());
31         }
32     }
33 }
```

```

1 namespace TeGe2
2 {
3     partial class Form1
4     {
5         /// <summary>
6         /// Variável de designer necessária.
7         /// </summary>
8         private System.ComponentModel.IContainer components = null;
9
10        /// <summary>
11        /// Limpar os recursos que estão sendo usados.
12        /// </summary>
13        /// <param name="disposing">true se for necessário descartar os recursos gerenciados; caso contrário, false.</param>
14        protected override void Dispose(bool disposing)
15        {
16            if (disposing && (components != null))
17            {
18                components.Dispose();
19            }
20            base.Dispose(disposing);
21        }
22
23        #region Código gerado pelo Windows Form Designer
24
25        /// <summary>
26        /// Método necessário para suporte ao Designer – não modifique
27        /// o conteúdo deste método com o editor de código.
28        /// </summary>
29        private void InitializeComponent()
30        {
31            this.components = new System.ComponentModel.Container();
32            this.dataGridView1 = new System.Windows.Forms.DataGridview();
33            this.label1 = new System.Windows.Forms.Label();
34            this.timer1 = new System.Windows.Forms.Timer(this.components);
35            this.label2 = new System.Windows.Forms.Label();
36            this.label3 = new System.Windows.Forms.Label();
37            this.dataGridView2 = new System.Windows.Forms.DataGridview();
38            this.dataGridView3 = new System.Windows.Forms.DataGridview();
39            this.label4 = new System.Windows.Forms.Label();
40            this.label5 = new System.Windows.Forms.Label();
41            this.label6 = new System.Windows.Forms.Label();
42            this.label7 = new System.Windows.Forms.Label();
43            this.label8 = new System.Windows.Forms.Label();
44            this.label9 = new System.Windows.Forms.Label();
45            this.label10 = new System.Windows.Forms.Label();
46            this.label11 = new System.Windows.Forms.Label();
47            this.label12 = new System.Windows.Forms.Label();
48            this.label13 = new System.Windows.Forms.Label();
49            this.label14 = new System.Windows.Forms.Label();
50            this.label15 = new System.Windows.Forms.Label();
51            this.label16 = new System.Windows.Forms.Label();
52            this.label17 = new System.Windows.Forms.Label();
53            ((System.ComponentModel.ISupportInitialize)(this.dataGridView1)).BeginInit();
54            ((System.ComponentModel.ISupportInitialize)(this.dataGridView2)).BeginInit();
55            ((System.ComponentModel.ISupportInitialize)(this.dataGridView3)).BeginInit();
56            this.SuspendLayout();
57            //
58            // dataGridView1
59            //
60            this.dataGridView1.AutoSizeColumnsMode = System.Windows.Forms.DataGridviewAutoSizeColumnsMode.Fill;
61            this.dataGridView1.BackgroundColor = System.Drawing.Color.Silver;
62            this.dataGridView1.ColumnHeadersHeightSizeMode = System.Windows.Forms.DataGridviewColumnHeadersHeightSizeMode.
        ↪ AutoSize;
63            this.dataGridView1.Location = new System.Drawing.Point(12, 38);
64            this.dataGridView1.Name = "dataGridView1";
65            this.dataGridView1.Size = new System.Drawing.Size(305, 243);
66            this.dataGridView1.TabIndex = 0;
67            //
68            // label1
69            //
70            this.label1.AutoSize = true;
71            this.label1.BackColor = System.Drawing.Color.White;
72            this.label1.ForeColor = System.Drawing.SystemColors.ControlText;
73            this.label1.Location = new System.Drawing.Point(125, 22);
74            this.label1.Name = "label1";
75            this.label1.Size = new System.Drawing.Size(71, 13);
76            this.label1.TabIndex = 1;
77            this.label1.Text = "Sala Principal";
78            //

```

```

79 // timer1
80 //
81 this.timer1.Enabled = true;
82 this.timer1.Interval = 200;
83 this.timer1.Tick += new System.EventHandler(this.timer1_Tick);
84 //
85 // label2
86 //
87 this.label2.AutoSize = true;
88 this.label2.Location = new System.Drawing.Point(451, 22);
89 this.label2.Name = "label2";
90 this.label2.Size = new System.Drawing.Size(91, 13);
91 this.label2.TabIndex = 2;
92 this.label2.Text = "Sala de Reuniões";
93 //
94 // label3
95 //
96 this.label3.AutoSize = true;
97 this.label3.Location = new System.Drawing.Point(785, 22);
98 this.label3.Name = "label3";
99 this.label3.Size = new System.Drawing.Size(91, 13);
100 this.label3.TabIndex = 3;
101 this.label3.Text = "Corredor de Baias";
102 //
103 // dataGridView2
104 //
105 this.dataGridView2.AutoSizeColumnsMode = System.Windows.Forms.DataGridViewAutoSizeColumnsMode.Fill;
106 this.dataGridView2.ColumnHeadersHeightSizeMode = System.Windows.Forms.DataGridViewAutoSizeColumnsMode.
    ↳ AutoSize;
107 this.dataGridView2.Location = new System.Drawing.Point(348, 38);
108 this.dataGridView2.Name = "dataGridView2";
109 this.dataGridView2.Size = new System.Drawing.Size(305, 243);
110 this.dataGridView2.TabIndex = 4;
111 //
112 // dataGridView3
113 //
114 this.dataGridView3.AutoSizeColumnsMode = System.Windows.Forms.DataGridViewAutoSizeColumnsMode.Fill;
115 this.dataGridView3.ColumnHeadersHeightSizeMode = System.Windows.Forms.DataGridViewAutoSizeColumnsMode.
    ↳ AutoSize;
116 this.dataGridView3.Location = new System.Drawing.Point(689, 38);
117 this.dataGridView3.Name = "dataGridView3";
118 this.dataGridView3.Size = new System.Drawing.Size(305, 243);
119 this.dataGridView3.TabIndex = 5;
120 //
121 // label4
122 //
123 this.label4.AutoSize = true;
124 this.label4.Location = new System.Drawing.Point(9, 311);
125 this.label4.Name = "label4";
126 this.label4.Size = new System.Drawing.Size(35, 13);
127 this.label4.TabIndex = 6;
128 this.label4.Text = "Relé: ";
129 //
130 // label5
131 //
132 this.label5.AutoSize = true;
133 this.label5.Location = new System.Drawing.Point(50, 311);
134 this.label5.Name = "label5";
135 this.label5.Size = new System.Drawing.Size(35, 13);
136 this.label5.TabIndex = 7;
137 this.label5.Text = "label5";
138 //
139 // label6
140 //
141 this.label6.AutoSize = true;
142 this.label6.Location = new System.Drawing.Point(386, 311);
143 this.label6.Name = "label6";
144 this.label6.Size = new System.Drawing.Size(35, 13);
145 this.label6.TabIndex = 9;
146 this.label6.Text = "label6";
147 //
148 // label7
149 //
150 this.label7.AutoSize = true;
151 this.label7.Location = new System.Drawing.Point(345, 311);
152 this.label7.Name = "label7";
153 this.label7.Size = new System.Drawing.Size(35, 13);
154 this.label7.TabIndex = 8;
155 this.label7.Text = "Relé: ";
156 //
157 // label8

```

```

158 //
159 this.label8.AutoSize = true;
160 this.label8.Location = new System.Drawing.Point(727, 311);
161 this.label8.Name = "label8";
162 this.label8.Size = new System.Drawing.Size(35, 13);
163 this.label8.TabIndex = 11;
164 this.label8.Text = "label8";
165 //
166 // label9
167 //
168 this.label9.AutoSize = true;
169 this.label9.Location = new System.Drawing.Point(686, 311);
170 this.label9.Name = "label9";
171 this.label9.Size = new System.Drawing.Size(35, 13);
172 this.label9.TabIndex = 10;
173 this.label9.Text = "Relé: ";
174 //
175 // label10
176 //
177 this.label10.AutoSize = true;
178 this.label10.Location = new System.Drawing.Point(12, 377);
179 this.label10.Name = "label10";
180 this.label10.Size = new System.Drawing.Size(173, 13);
181 this.label10.TabIndex = 12;
182 this.label10.Text = "Total de pessoas dentro do LARA: ";
183 //
184 // label11
185 //
186 this.label11.AutoSize = true;
187 this.label11.Location = new System.Drawing.Point(191, 377);
188 this.label11.Name = "label11";
189 this.label11.Size = new System.Drawing.Size(41, 13);
190 this.label11.TabIndex = 13;
191 this.label11.Text = "label11";
192 //
193 // label12
194 //
195 this.label12.AutoSize = true;
196 this.label12.Location = new System.Drawing.Point(345, 284);
197 this.label12.Name = "label12";
198 this.label12.Size = new System.Drawing.Size(180, 13);
199 this.label12.TabIndex = 14;
200 this.label12.Text = "Indicador para o acionameto do relé:";
201 //
202 // label13
203 //
204 this.label13.AutoSize = true;
205 this.label13.Location = new System.Drawing.Point(9, 284);
206 this.label13.Name = "label13";
207 this.label13.Size = new System.Drawing.Size(180, 13);
208 this.label13.TabIndex = 15;
209 this.label13.Text = "Indicador para o acionameto do relé:";
210 //
211 // label14
212 //
213 this.label14.AutoSize = true;
214 this.label14.Location = new System.Drawing.Point(686, 284);
215 this.label14.Name = "label14";
216 this.label14.Size = new System.Drawing.Size(180, 13);
217 this.label14.TabIndex = 16;
218 this.label14.Text = "Indicador para o acionameto do relé:";
219 //
220 // label15
221 //
222 this.label15.AutoSize = true;
223 this.label15.Location = new System.Drawing.Point(195, 284);
224 this.label15.Name = "label15";
225 this.label15.Size = new System.Drawing.Size(41, 13);
226 this.label15.TabIndex = 17;
227 this.label15.Text = "label15";
228 //
229 // label16
230 //
231 this.label16.AutoSize = true;
232 this.label16.Location = new System.Drawing.Point(531, 284);
233 this.label16.Name = "label16";
234 this.label16.Size = new System.Drawing.Size(41, 13);
235 this.label16.TabIndex = 18;
236 this.label16.Text = "label16";
237 //
238 // label17

```

```

239 //
240 this.label17.AutoSize = true;
241 this.label17.Location = new System.Drawing.Point(872, 284);
242 this.label17.Name = "label17";
243 this.label17.Size = new System.Drawing.Size(41, 13);
244 this.label17.TabIndex = 19;
245 this.label17.Text = "label17";
246 //
247 // Form1
248 //
249 this.AutoScaleDimensions = new System.Drawing.SizeF(6F, 13F);
250 this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;
251 this.BackColor = System.Drawing.Color.White;
252 this.ClientSize = new System.Drawing.Size(1006, 399);
253 this.Controls.Add(this.label17);
254 this.Controls.Add(this.label16);
255 this.Controls.Add(this.label15);
256 this.Controls.Add(this.label14);
257 this.Controls.Add(this.label13);
258 this.Controls.Add(this.label12);
259 this.Controls.Add(this.label11);
260 this.Controls.Add(this.label10);
261 this.Controls.Add(this.label8);
262 this.Controls.Add(this.label9);
263 this.Controls.Add(this.label6);
264 this.Controls.Add(this.label7);
265 this.Controls.Add(this.label5);
266 this.Controls.Add(this.label4);
267 this.Controls.Add(this.dataGridView3);
268 this.Controls.Add(this.dataGridView2);
269 this.Controls.Add(this.label3);
270 this.Controls.Add(this.label2);
271 this.Controls.Add(this.label1);
272 this.Controls.Add(this.dataGridView1);
273 this.Name = "Form1";
274 this.Text = "Interface Gráfica TG2 - Leonardo e Kenneth";
275 this.Load += new System.EventHandler(this.Form1_Load);
276 ((System.ComponentModel.ISupportInitialize)(this.dataGridView1)).EndInit();
277 ((System.ComponentModel.ISupportInitialize)(this.dataGridView2)).EndInit();
278 ((System.ComponentModel.ISupportInitialize)(this.dataGridView3)).EndInit();
279 this.ResumeLayout(false);
280 this.PerformLayout();
281
282 }
283
284 #endregion
285
286 private System.Windows.Forms.DataGridView dataGridView1;
287 private System.Windows.Forms.Label label1;
288 private System.Windows.Forms.Timer timer1;
289 private System.Windows.Forms.Label label2;
290 private System.Windows.Forms.Label label3;
291 private System.Windows.Forms.DataGridView dataGridView2;
292 private System.Windows.Forms.DataGridView dataGridView3;
293 private System.Windows.Forms.Label label4;
294 private System.Windows.Forms.Label label5;
295 private System.Windows.Forms.Label label6;
296 private System.Windows.Forms.Label label7;
297 private System.Windows.Forms.Label label8;
298 private System.Windows.Forms.Label label9;
299 private System.Windows.Forms.Label label10;
300 private System.Windows.Forms.Label label11;
301 private System.Windows.Forms.Label label12;
302 private System.Windows.Forms.Label label13;
303 private System.Windows.Forms.Label label14;
304 private System.Windows.Forms.Label label15;
305 private System.Windows.Forms.Label label16;
306 private System.Windows.Forms.Label label17;
307 }
308 }

```

Abaixo se encontra o código utilizado no arduino para o controle dos relés. O programa foi feito na IDE do Arduino.

ArduinoSerial.ino

```
1  const byte numChars = 32;
2  char receivedChars[numChars];
3  char tempChars[numChars]; // vetor temporario para leitura dos dados seriais
4
5      // variaveis que guardam os estados lidos
6  char SALA[numChars] = {};
7  //char SALA;
8  int N_SALA;
9  int ESTADO = 0;
10 float floatFromPC = 0.0;
11
12 boolean newData = false;
13
14 int LED_SALA_DE_REUNIOES = 2;
15 int LED_SALA_PRINCIPAL = 3;
16 int LED_CORREDOR_DE_BAIAS = 4;
17
18 //=====
19
20 void setup() {
21     Serial.begin(9600);
22     //Serial.println("Formato do dado: <HelloWorld, 12, 24.7> ");
23     //Serial.println();
24 }
25
26 //=====
27
28 void loop() {
29     rcvWithStartEndMarkers();
30     if (newData == true) {
31         strcpy(tempChars, receivedChars);
32         parseData();
33         //showParsedData();
34         correspondencia();
35         ativa_rele();
36         newData = false;
37     }
38 }
39
40 //=====
41
42 void rcvWithStartEndMarkers() {
43     static boolean rcvInProgress = false;
44     static byte ndx = 0;
45     char startMarker = '<';
46     char endMarker = '>';
47     char rc;
48
49     while (Serial.available() > 0 && newData == false) {
50         rc = Serial.read();
51
52         if (rcvInProgress == true) {
53             if (rc != endMarker) {
54                 receivedChars[ndx] = rc;
55                 ndx++;
56                 if (ndx >= numChars) {
57                     ndx = numChars - 1;
58                 }
59             }
60             else {
61                 receivedChars[ndx] = '\0'; // coloca o nulo no fim do vetor
62                 rcvInProgress = false;
63                 ndx = 0;
64                 newData = true;
65             }
66         }
67         else if (rc == startMarker) {
68             rcvInProgress = true;
69         }
70     }
71 }
72
73 //=====
74
75
```

```

76 void parseData() { // faz o parse
77   char * strtokIndx;
78
79   strtokIndx = strtok(tempChars, ","); // pega o primeiro estado que vem antes da virgula
80   strcpy(SALA, strtokIndx);
81
82   strtokIndx = strtok(NULL, ","); // pega o segundo estado, depois da virgula
83   ESTADO = atoi(strtokIndx);
84
85   //strtokIndx = strtok(NULL, ",");
86   //floatFromPC = atof(strtokIndx);
87
88 }
89
90 //=====
91
92 void showParsedData() {
93   Serial.print("SALA: ");
94   Serial.println(SALA);
95   Serial.print("ESTADO: ");
96   Serial.println(ESTADO);
97   //Serial.print("Float ");
98   //Serial.println(floatFromPC);
99 }
100
101 void correspondencia(){
102   if ((SALA[0] == 'R')||(SALA[0] == 'r')){
103     //Serial.println("ENTROU R");
104     N_SALA = 0;}
105   else if ((SALA[0] == 'P')||(SALA[0] == 'p')){
106     //Serial.println("ENTROU P");
107     N_SALA = 1;}
108   else if ((SALA[0] == 'C')||(SALA[0] == 'c')){
109     //Serial.println("ENTROU C");
110     N_SALA = 2;}
111   //Serial.println("\nSALA DA CORRESPONDENCIA");
112   //Serial.print("-");
113   //Serial.print(SALA);
114   //Serial.println("-");
115   //Serial.println("N_SALA:");
116   //Serial.println(N_SALA);
117   //Serial.println();
118 }
119
120 void ativa_rele(){
121   switch(N_SALA){
122     case 0:
123       //Serial.println("ATIVO SALA REUNIÕES");
124       //estado(ESTADO);
125       digitalWrite(LED_SALA_DE_REUNIOES,!ESTADO);
126       break;
127
128     case 1:
129       //Serial.println("ATIVO SALA PRINCIPAL");
130       //estado(ESTADO);
131       digitalWrite(LED_SALA_PRINCIPAL,!ESTADO);
132       break;
133
134     case 2:
135       //Serial.println("ATIVO CORREDOR DE BAIAS");
136       //estado(ESTADO);
137       digitalWrite(LED_CORREDOR_DE_BAIAS,!ESTADO);
138       break;
139
140     default:
141       //Serial.println("CASO NÃO IDENTIFICADO");
142       break;
143   }
144 }
145
146 void estado(int estado){
147   if (estado == 0)
148     Serial.println("DESLIGADO");
149   else
150     Serial.println("LIGADO");
151 }

```

II. MANUAL DE INSTALAÇÃO E CONFIGURAÇÃO

II.1 OpenHAB

OpenHAB (Open Home Automation Bus) é uma plataforma open-source feita para automação residencial e predial.

II.1.1 Pre-requisitos

- Raspberry Pi 2 ou mais novo.
- Cartão SD de 16G ou mais.
- Fonte de alimentação estável.
- Conexão com internet.
- Java 8 (Recomendado: Zulu).

II.1.2 Instalação OpenHAB

Neste manual de instalação assume-se que o usuário possui um raspberry pi com Raspbian instalado.

Passo 1: adicionar chave do repositório Bintray do OpenHAB no gerenciador de pacotes e permitir que Apt use o protocolo HTTPS. Comandos:

```
1 $ wget -qO - 'https://bintray.com/user/downloadSubjectPublicKey?username=openhab' |  
  ↪ sudo apt-key add -
```

```
1 $ sudo apt-get install apt-transport-https
```

Passo 2: adicionar repositório estável para a lista Apt

```
1 $ echo 'deb https://dl.bintray.com/openhab/apt-repo2 stable main' | sudo tee \newline/etc  
  ↪ /apt/sources.list.d/openhab2.list
```

Passo 3: sincronizar novamente o index de pacotes

```
1 sudo apt-get update
```

Passo 4: instalar OpanHAB

```
1 $ sudo apt-get install openhab2
```

Passo 5: instalar pacote de add-ons

```
1 $ sudo apt-get install openhab2-addons
```

II.1.3 Instalação Mosquitto MQTT Broker

II.1.3.1 Instalação do Mosquitto MQTT Broker por meio do repositório Debian

Importar e instalar chave de repositório:

```
1 $ wget http://repo.mosquitto.org/debian/mosquitto-repo.gpg.key
```

```
1 $ sudo apt-key add mosquitto-repo.gpg.key
```

Tornar o repositório disponível para o Apt:

```
1 $ cd /etc/apt/sources.list.d/
```

Use apenas um dos seguintes comandos dependendo da versão do Raspbian que o usuário estiver usando

```
1 $ sudo wget http://repo.mosquitto.org/debian/mosquitto-wheezy.list
```

```
1 $ sudo wget http://repo.mosquitto.org/debian/mosquitto-jessie.list
```

```
1 $ sudo wget http://repo.mosquitto.org/debian/mosquitto-stretch.list
```

Atualize

```
1 sudo apt-get update
```

Instale Mosquitto

```
1 $ sudo apt-get install mosquitto
```

II.1.3.2 Instalação do Mosquitto MQTT Broker por meio do código fonte

O tutorial explicado acima é a forma recomendada por ser o mais fácil. No entanto não funcionou durante a instalação do Mosquitto no Raspberry Pi utilizado neste projeto. O processo utilizado foi a instalação por meio do código fonte

Atualização de pacotes:

```
1 $ sudo apt-get update
2 $ sudo apt-get dist-upgrade
```

Instalação de dependências:

```
1 $ sudo apt-get install cmake libssl-dev libwebsockets-dev uuid-dev libc-ares-dev
```

Crie uma pasta temporária para usar durante a instalação e para os arquivos compilados. Depois entre na pasta.

```
1 $ mkdir temp
2 $ cd temp
```

Faça o download dos arquivos do código fonte e descompacte para c-ares

```
1 $ wget http://c-ares.haxx.se/download/c-ares-1.12.0.tar.gz
2 $ tar -xvf c-ares-1.12.0.tar.gz
```

Entre no diretório c-ares

```
1 $ cd c-ares-1.12.0
```

Configurar e compilar c-ares

```
1 $ ./configure
2 $ make -j4
```

Instalar c-ares:

```
1 $ sudo make install
```

Volte ao diretório temporário

```
1 $ cd ..
```

Faça o download dos arquivos do código fonte de Mosquitto e descompacte

```
1 $ wget http://mosquitto.org/files/source/mosquitto-1.4.15.tar.gz
2 tar xavf mosquitto-1.4.15.tar.gz
```

Entre no diretório do Mosquitto

```
1 $ cd mosquitto-1.4.15
```

Execute o comando cmake com a opção websockets

```
1 $ cmake -DWITH_WEBSOCKETS=ON .
```

Execute o arquivo compilado e instale Mosquitto

```
1 $ make -j4
2 $ sudo make install
```

Volte ao diretório principal

```
1 $ cd ~
```

Remova o diretório temporário para liberar espaço

```
1 $ rm -r temp
```

Adicione Mosquitto como usuário

```
1 $ useradd -r -m -d /var/lib/mosquitto -s /usr/sbin/nologin -g nogroup mosquitto
```

II.1.3.3 Habilitando Websockets

Nesta etapa precisamos editar o arquivo de configuração do Mosquitto. Dependendo do método de instalação do Mosquitto utilizado o arquivo se encontra em locais diferentes.

Se a instalação foi feita por meio do código fonte, o arquivo se encontra em: **/usr/local/etc/mosquitto/mosquitto.conf**

Se a instalação foi feita por meio do repositório debian, o arquivo se encontra em: **/usr/share/doc/mosquitto/examples/mosquitto.conf.gz**

O arquivo de configuração pode estar compactado. Para descompactar utilize os comandos:

```
1 $ cd /usr/share/doc/mosquitto/examples
2 $ gunzip mosquitto.conf.gz
```

II.1.3.4 Editando o arquivo mosquitto.conf

Para editar o arquivo mosquitto.conf pode ser utilizado qualquer editor de texto. O editor usado neste projeto foi o nano. Portanto nos comandos a seguir vão considerar nano como o editor a ser utilizado.

Abra o arquivo mosquitto.conf no editor de texto. Caso esteja no mesmo diretório do arquivo

```
1 $ sudo nano mosquitto.conf
```

Caso esteja em qualquer outro diretório

```
1 $ sudo editor /usr/local/etc/mosquitto/mosquitto.conf
```

Atualize as seguintes linhas do código (a maior parte das atualizações se dão apenas pelo remoção do # para descomentar a linha):

```
1 linha 35: pid_file /var/run/mosquitto.pid
2 linha 43: user mosquitto
3 linha 133: bind_address 0.0.0.0
4 linha 136: port 1883
5 linha 151: protocol mqtt
```

Copie o arquivo config editado

```
1 $ mv /etc/mosquitto/mosquitto.conf /etc/mosquitto/mosquitto.conf.bak
```

Depois entre no diretório

```
1 $ cp mosquitto.conf /etc/mosquitto/
```

Teste a instalação

```
1 $ mosquitto -c /etc/mosquitto/mosquitto.conf
```

O resultado deve ser algo parecido com

```
1 1512012962: mosquitto version 1.4.14 (build date 2017-11-29 15:51:45-0800) starting
2 1512012962: Config loaded from /usr/local/etc/mosquitto/mosquitto.conf.
3 1512012962: Opening websockets listen socket on port 9001.
4 1512012962: Opening ipv4 listen socket on port 1883.
```

II.2 Arduino

II.2.1 ESP8266

O ESP8266 é um chip que permite o envio de dados entre diversos dispositivos utilizando WiFi. Existem diversos módulos e placas de desenvolvimento de várias fabricantes e o que é utilizado neste trabalho é o módulo esp8266 ESP-01.

O módulo ESP 01 é a versão mais comum do ESP8266 e também o mais básico. Ele pode operar por meio da utilização de comandos "AT" ou pode operar como standalone, onde ele trabalha como um microcontrolador por si só.

Para o nosso projeto o módulo ESP 01 será operado por meio de comandos AT, sendo conectado diretamente num Arduino. Para que possamos fazer essa conexão precisamos primeiro conhecer a pinagem do módulo, e isso está ilustrado na figura II.1.

II.2.1.1 Configuração do módulo ESP-01

Antes de utilizarmos o módulo ESP8266 ESP-01 precisamos primeiro atualizar o seu firmware e mudar seu baudrate. Para isso utilizamos os programas:

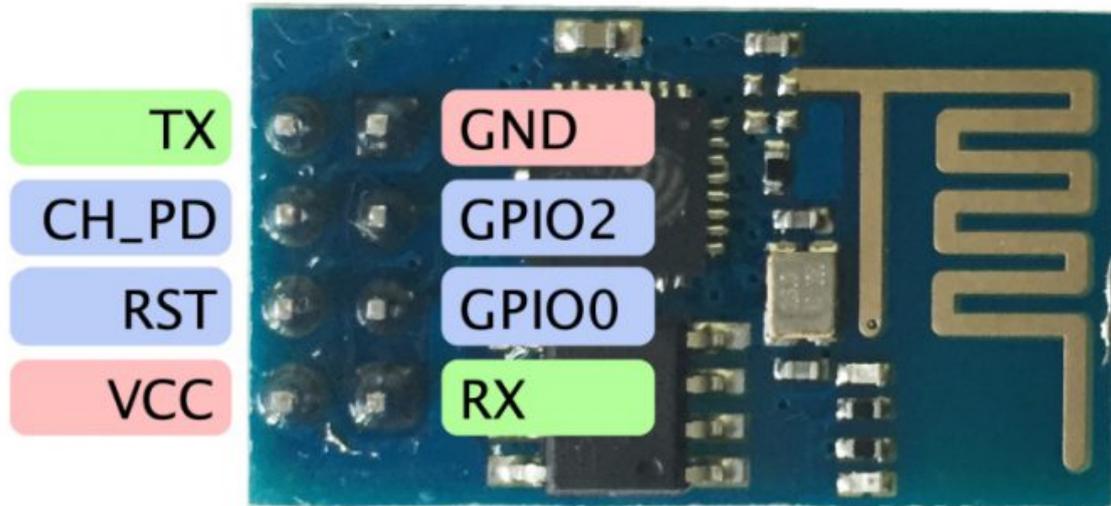


Figura II.1: Pinagem do módulo ESP-01

- Termit
- Flash Downloads Tools
- Arduino IDE

Antes de utilizar esses programas é preciso saber que o modo de conexão dos pinos do módulo muda de acordo com o seu objeto, que pode ser o modo de uso normal ou pode ser o modo de programação do módulo. Será explicado um desses modos de cada vez, à medida que formos avançando no projeto.

Primeiro precisamos fazer o upgrade do firmware que vem no ESP-01. Para isso é necessário seguir dois passos.

II.2.1.2 Preparação para upgrade do firmware

PASSO 1

Carregue o seguinte código para o Arduino para garantir que a porta serial não esteja ocupada.

```

1 void setup()
2   {
3     //vazio
4   }
5 void loop()
6   {
7     //vazio
8   }

```

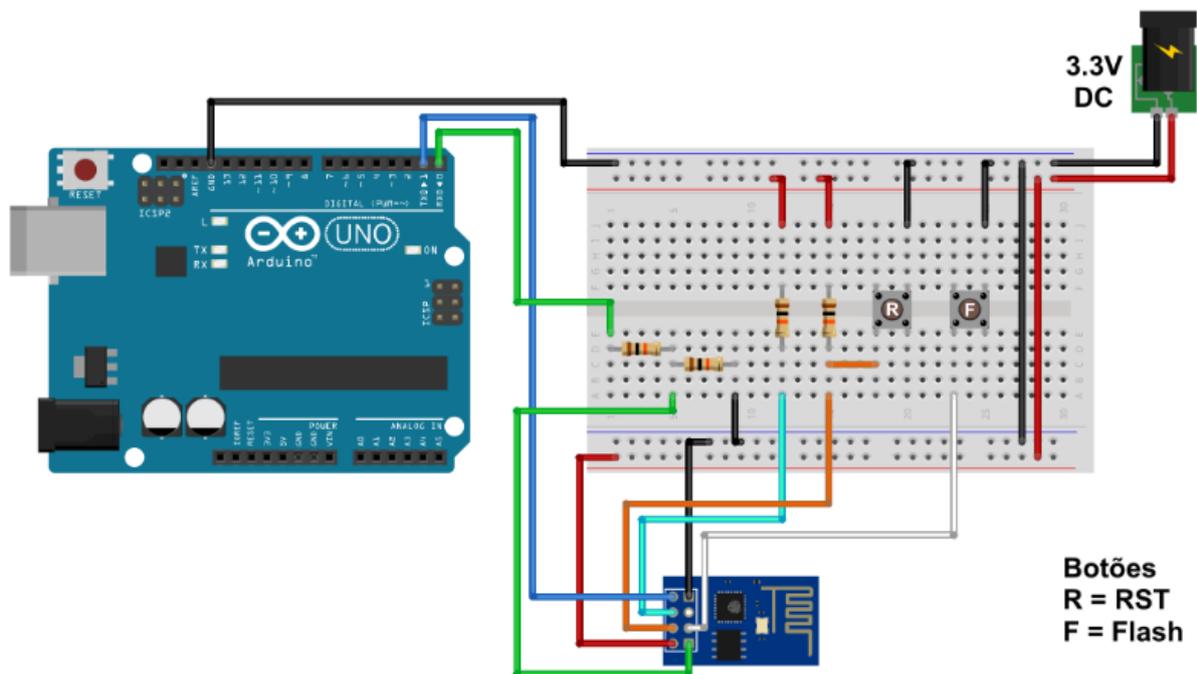


Figura II.2: Circuito e conexão dos pinos do módulo ESP-01 para modo programação

PASSO 2

Monte o circuito para deixar o módulo ESP-01 no modo de programação. O circuito e a conexão dos pinos do módulo devem ser feitas de acordo com a figura II.2.

II.2.1.3 Verificação da versão do firmware

Para verificar a versão do firmware que vem no ESP-01 utilizamos o programa Termitte.

Antes de utilizar esse programa precisamos modificar as configurações na aba "settings". Ela deve estar de acordo com a figura II.3.

Na configuração, a opção da "porta" deve ser selecionada de acordo com a porta na qual o arduino está conectada. Esta pode ser verificada usando o IDE do arduino.

Feito as configurações necessárias no Termitte, clique no botão "click to connect" para conectar. Depois digite o comando

1 AT

na parte inferior do programa. Se a resposta for "OK", significa que estamos conseguindo nos comunicar com o módulo. Depois digite o comando

1 AT+GMR

Isso está ilustrado na figura II.4.

Serial port settings

Port configuration

Port: COM3

Baud rate: 115200

Data bits: 8

Stop bits: 1

Parity: none

Flow control: none

Forward: none

Transmitted text

Append nothing

Append CR

Append LF

Append CR-LF

Local echo

Received text

Polling: 100 ms

Max. lines:

Font: default

Word wrap

Options

Stay on top

Quit on Escape

Autocomplete edit line

Keep history

Close port when inactive

Plug-ins

Auto Reply

Function Keys

Hex View

Highlight

Log File

User interface language: English (en)

Cancel OK

Figura II.3: Configuração do programa Termit

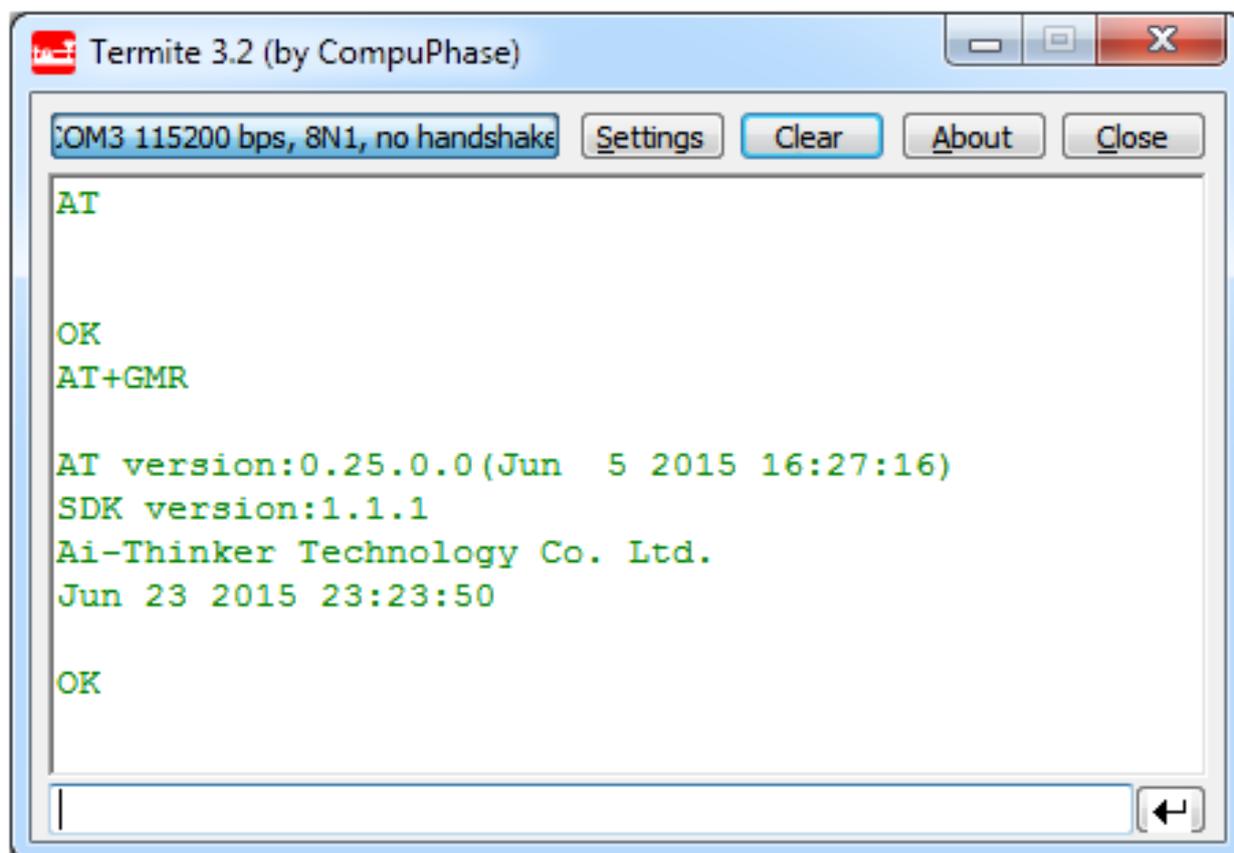


Figura II.4: Comando e resposta do programa Termit

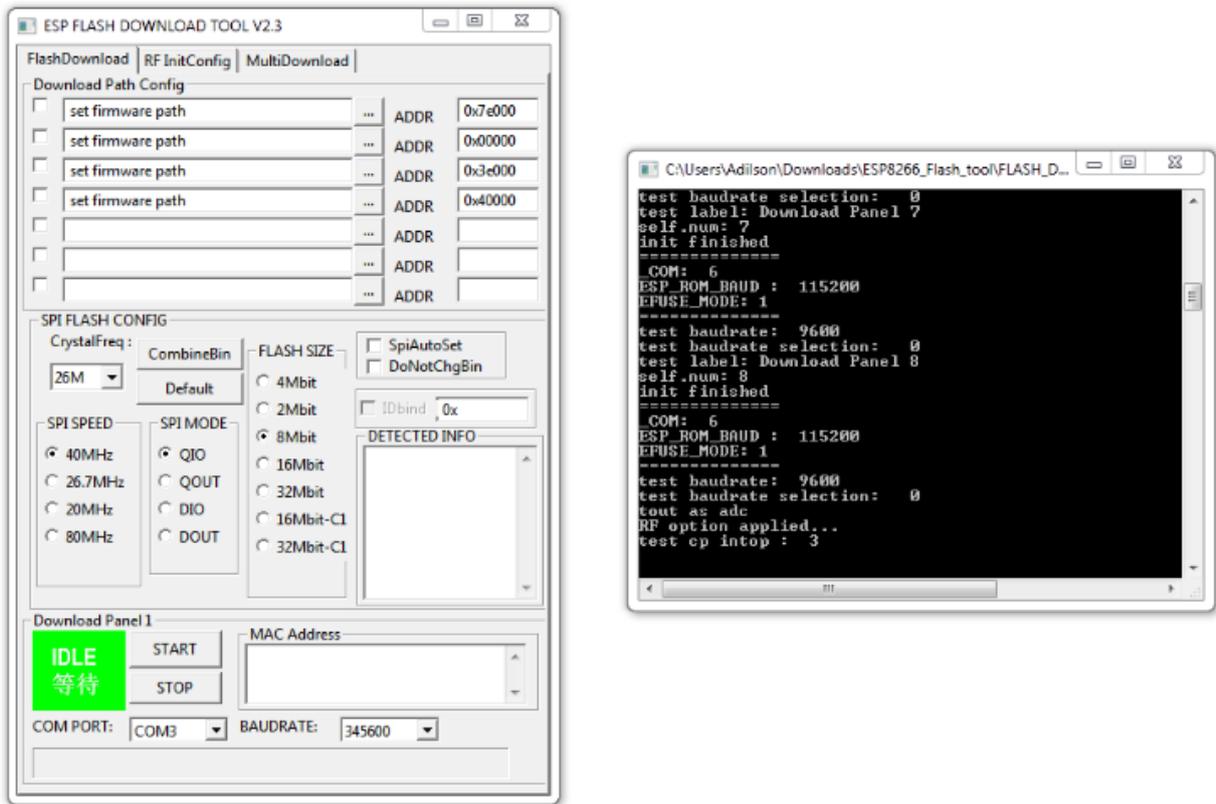


Figura II.5: Janelas que abrem ao executar ESP Flash Download Tool

O exemplo da figura II.4 mostra que o firmware está na versão 1.1.1 e os comandos AT estão na versão 0.25.0.0.

Agora sabemos a versão do firmware que está no módulo ESP-01. Com isso o usuário pode decidir se usará esta versão ou se prefere usar outra. Neste projeto escolhemos usar a versão 1.5.0 porque encontramos uma quantidade maior de tutoriais para o módulo ESP-01 que utilizavam esta versão.

Para atualizar a versão do firmware, utilizamos o programa ESP Flash Download Tool.

II.2.1.4 Atualização da versão do firmware

O upgrade da versão do firmware será feito usando o programa ESP Flash Download Tool. O seu download pode ser feito pelo link:

1 <http://bbs.espressif.com/viewtopic.php?f=57&t=433>

Ao executar o programa será aberto duas janelas. Uma é a janela principal onde serão feitas as configurações. A outra é dos logs.

As janelas que aparecem estão ilustradas na figura II.5

Para a configuração do programa, mude a opção da "COM PORT", que se encontra na parte

inferior da janela principal, para a porta na qual está conectada o arduino.

Agora tome como base o circuito montado na seção II.2.1.2, figura II.2. Neste circuito há dois botões: RST (esquerda) e Flash (direita). Para fazer com que o módulo ESP-01 entre no modo programação, mantenha pressionado o botão RST e em seguida pressione o botão Flash. Então solte o botão RST e depois o botão Flash. Agora que o módulo está no modo de programação, volte ao programa e acione o botão START. Com isso as informações contidas no módulo serão lidas.

Agora é o momento de passar a versão desejada do firmware para o módulo. Se também deseja usar a versão que usamos, pode baixá-lo no link abaixo:

```
1 http://bbs.espressif.com/viewtopic.php?f=46&t=1442#p4795
```

Com o download do firmware da versão que desejamos feito, precisamos selecioná-lo no programa. Para isso, no programa selecione a opção SpiAutoSet e selecione os quatro arquivos abaixo em "Download Path Config":

```
1 ESP8266_Flash_toolFirmwareesp_iot_sdk_v1.5.0binesp_init_data_default.bin
2
3 ESP8266_Flash_toolFirmwareesp_iot_sdk_v1.5.0binblank.bin
4
5 ESP8266_Flash_toolFirmwareesp_iot_sdk_v1.5.0binboot_v1.4(b1).bin
6
7 ESP8266_Flash_toolFirmwareesp_iot_sdk_v1.5.0binat512+512user1.1024.new.2.bin
```

Depois preencha os endereços ADDR de acordo com a figura II.6.

Acione o botão START para que o upgrade seja iniciado. Ao fim do processo, a janela de log deve mostrar as informações ilustradas na figura II.7.

Agora que o módulo está com a versão do firmware que desejamos, reinicie-o desligando e ligando novamente.

II.2.1.5 Setando baudrate

O módulo ESP-01 vem de fábrica com o baudrate setado em 115200. Neste projeto trabalharemos com ele com o baudrate de 9600. Para fazer essa mudança, feche o programa ESP Flash Download Tool, carregue o programa descrito na seção II.2.1.2 para o arduino e execute novamente o programa Termite.

Digite novamente o comando

```
1 AT+GMR
```

para confirmar que estamos com a versão de firmware desejada. A resposta deve ser a seguinte:

```
1 AT version: 0.51.0.0 (Nov 27 2015 13:37:21)
```

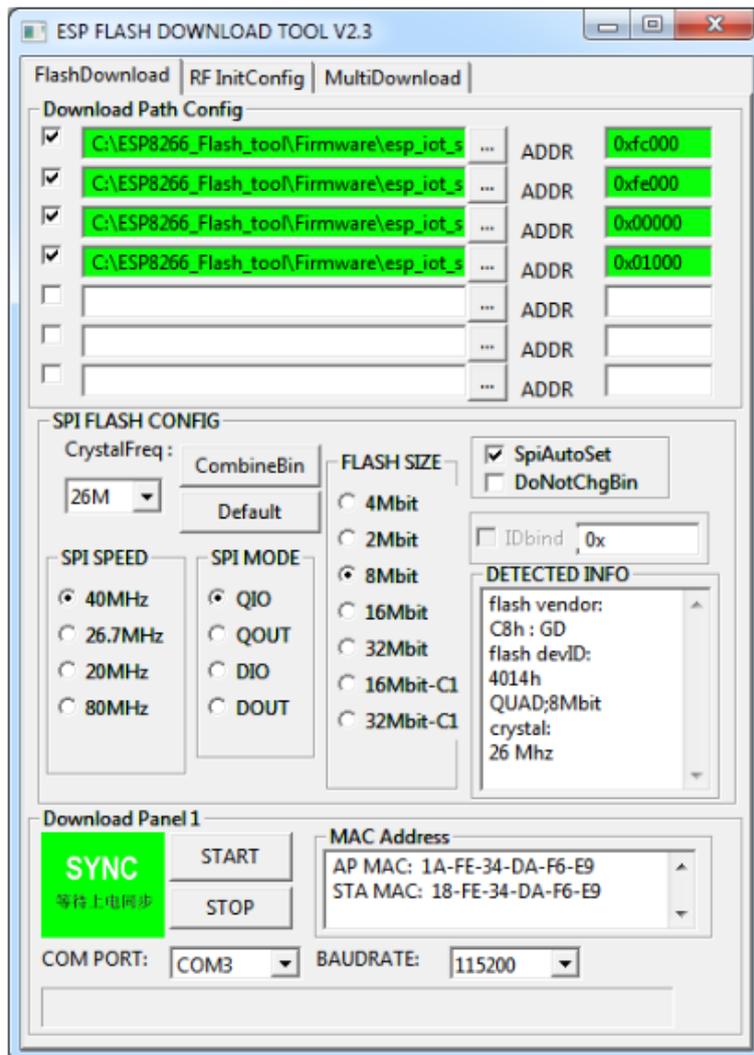


Figura II.6: Endereços ADDR ao lado de seus respectivos arquivos

```
offset : 1040384
Erasing flash...
head: 1 ;total: 1
erase size : 4096
Writing at 0x00000000... <0 %> mode: 0
size : 2
sel.freq: 26000000
ttttest crystal : 1
Writing at 0x00000800... <0 %>
Leaving...
Erasing flash...
head: 15 ;total: 70
erase size : 225280
Writing at 0x00046000... <97 %>
Leaving...
Erasing flash...
head: 1 ;total: 1
erase size : 4096
Writing at 0x000fc000... <97 %>
Leaving...
Erasing flash...
head: 1 ;total: 1
erase size : 4096
Writing at 0x000fec00... <99 %>
Leaving...
com closed
```

Figura II.7: Resultado na janela log ao fim do upgrade

2 | SDK version:1.5.0

Confirmado a versão, vamos mudar o baudrate para 9600. Para fazer isso utilize o comando:

1 | AT+UART_DEF=9600,8,1,0,0

Com isso terminamos a configuração necessária do módulo ESP-01.

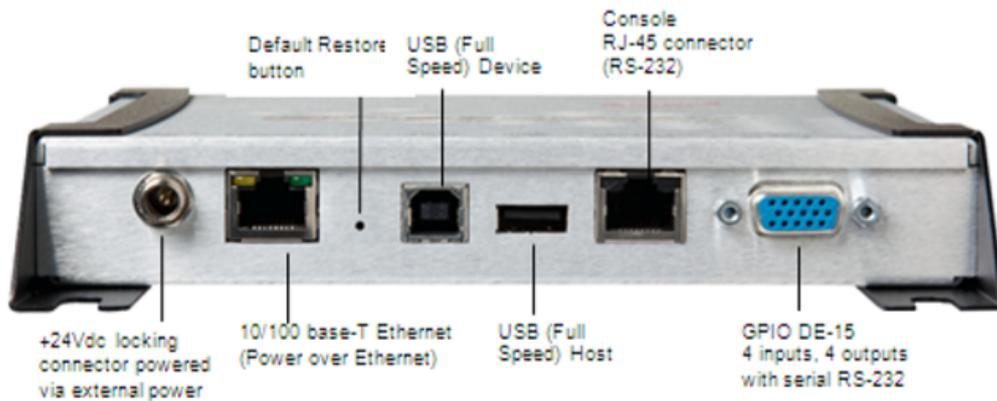


Figura II.8: Portas E/S localizadas no Reader Speedway R420. Imagem retirada de [3].

II.3 Instalação e teste das leitoras e antenas

O guia da instalação e teste das leitoras e antenas a seguir se concentra na instalação e operação de apenas uma leitora para simplificação. Repita os procedimentos para quantas leitoras forem necessárias.

II.3.1 Instalação física

A primeira etapa para a utilização do sistema é fazer a instalação física das leitoras e antenas. As etapas a seguir foram retiradas e adaptadas do manual de instalação e operação da Impinj SpeedwayR [3].

Requisitos para usar o Speedway:

- Temperatura de operação: -20°C a $+ 50^{\circ}\text{C}$ (sem condensação).
- Equipamento de rede TCP/IP para conectar o Speedway Reader a um computador (Windows, Mac ou Linux) ou outro terminal de rede.
- A conexão à porta do console do Speedway Reader requer um cabo de rede (conector RJ-45 para DB9) e uma porta serial RS-232 ou um adaptador serial para USB no computador.
- Antena UHF RFID aprovada por Impinj, incluindo cabos RF associados ou cabos com uma interface de conector macho RP-TNC.

A figura II.8 ilustra as portas de E/S localizadas no Speedway Reader. A leitora utilizada é uma Speedway R420, que inclui quatro portas de antena.

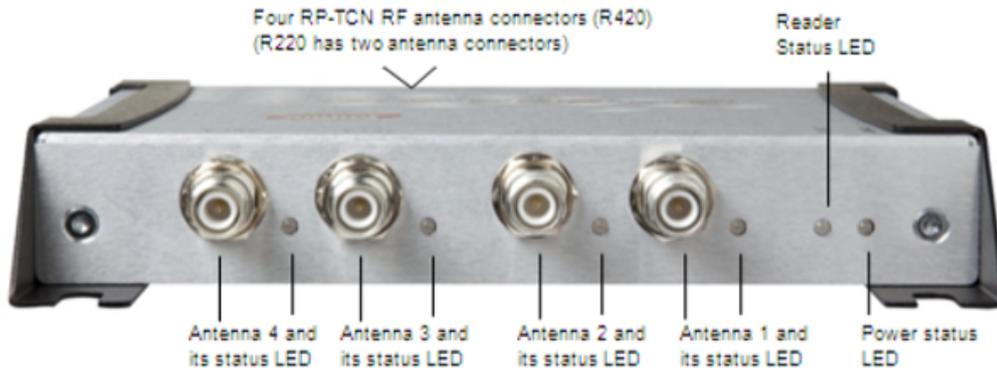


Figura II.9: Portas da antena e LEDs no Reader Speedway R420. Imagem retirada de [3].

As portas da antena e os indicadores de status do LED estão localizados no painel traseiro do Reader. A figura II.9 da Speedway R420 ilustra seus locais.

II.3.1.1 Padrões de piscamento dos LEDs Speedway Reader

O Speedway Reader possui vários LEDs para indicar o status operacional do Reader. As três categorias principais de LED são energia, status do Reader e status da antena. Cada LED possui seus próprios padrões para transmitir status ao usuário.

A tabela II.3.1.1 define os padrões para o LED de Power.

LED State	Reader State
Solid RED (after power-on or reset)	Power applied, attempting to start boot code
OFF	Default Restore button pressed
One short RED blink	Configuration Default Restore detected
Two short RED blinks	Factory Default Restore detected
Blinking RED (4 Hz)	Unable to boot (see console for details)
Solid GREEN	Done booting, starting application image
Blinking ORANGE (1Hz)	USB flash drive upgrade in progress
Blinking RED (2 Hz)	USB flash drive upgrade failure

Tabela II.1: Padrões do LED de power. Informações retiradas de [3].

A tabela II.3.1.1 define os padrões para o LED do status do Reader.

LED State	Reader State
OFF	Application image booting, RFID not available
Alternating RED and GREEN	Application image booting, RFID not available
Solid GREEN	Application image booted, RFID available, No LLRP connection
Two short GREEN blinks	Active LLRP connection
One short GREEN blink	No LLRP connection, LLRP ROSpec configured and enabled
Blinking ORANGE	Inventory active, blinking rate increases with number of tags

Tabela II.2: Padrões do LED do status do Reader. Informações retiradas de [3].

A tabela II.3.1.1 define os padrões para o LED da antena.

LED State	Reader State
OFF	Application image booting, RFID not available
Alternating RED and GREEN	Application image booting, RFID not available

Tabela II.3: Padrões do LED do status do antena. Informações retiradas de [3].

As etapas principais de instalação e conexão do Speedway são:

- Posicione o Reader adequadamente no seu ambiente.
- Conecte as antenas às portas apropriadas do Reader.
- Conecte a fonte ao Reader.
- Conecte o Reader à rede.
- Configure a opção de região do Reader.
- Teste a instalação do Reader lendo tags.



Figura II.10: Exemplo de rede configuração de rede utilizando cabo ethernet e switch de rede. Imagem retirada de [3].

II.3.2 Configuração da rede

Após a devida instalação física do Reader, deve-se configurar a rede apropriadamente. No caso deste trabalho foi utilizado um cabo Ethernet em um switch para fazer uma rede local fechada e um computador com sistema operacional Windows 10.

O primeiro passo é configurar o computador com IP fixo. Para isso, acesse as propriedades de Wi-Fi e clique duas vezes na opção "Protocolo IP Versão 4 (TCP/IPv4)", como é mostrado na figura II.11.

Dentro das propriedades do protocolo IPv4, selecione a opção "Usar o seguinte endereço IP:" e preencha os campos, vide figura II.12.

Em seguida, faça o download do Bonjour Print Service para Windows. A partir de agora deveria ser possível acessar os Readers que estão conectados na rede. Desta forma, faça um teste abrindo o prompt e utilizando a função "ping" nos Readers que estão conectados na rede. Para isso, digite "ping speedwayr-XX-XX-XX.local", de modo que "XX-XX-XX" são os últimos três bytes do endereço MAC do Reader (impresso na etiqueta da versão anexada ao estojo do Reader). Neste exemplo os últimos três bytes do Reader utilizado são "10-9F-BB", como é mostrado na figura II.13.

Caso todos os pacotes enviados tenham sido recebidos, quer dizer que a rede foi configurada corretamente. Em seguida, o próximo passo é acessar o IP do Reader. Para isso, acesse o site <http://speedwayr-XX-XX-XX>, de modo que "XX-XX-XX" são os últimos três bytes do endereço MAC do Reader, e coloque o nome de usuário "root" e senha "impinj", como é mostrado na figura II.14.

Se tudo estiver correto, será possível acessar a interface web do Reader. Nessa interface é possível visualizar informações importantes do Reader, como por exemplo o endereço IP. A figura II.15 mostra a tela da interface web do Reader utilizado.

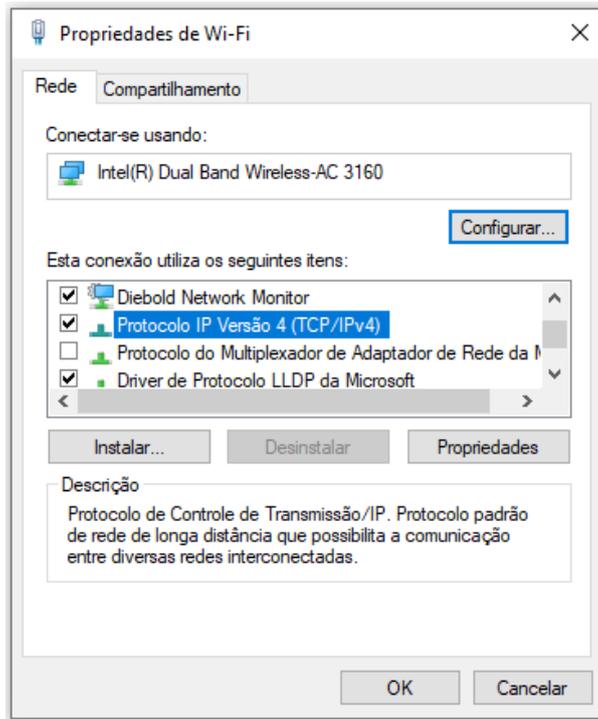


Figura II.11: Configurações da rede Wi-Fi.

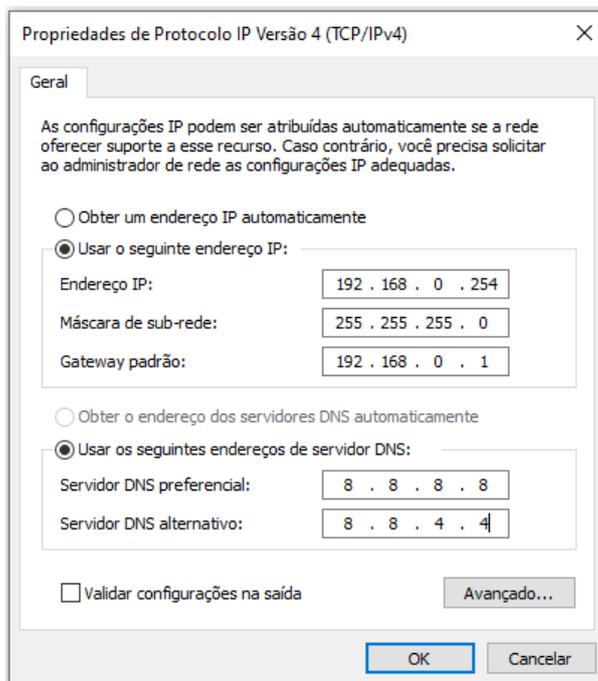


Figura II.12: Configuração do IP fixo.

```
CA Prompt de Comando
Microsoft Windows [versão 10.0.18363.836]
(c) 2019 Microsoft Corporation. Todos os direitos reservados.

C:\Users\leona>ping speedwayr-10-9F-BB.local

Disparando speedwayr-10-9F-BB.local [192.168.0.241] com 32 bytes de dados:
Resposta de 192.168.0.241: bytes=32 tempo=1ms TTL=64
Resposta de 192.168.0.241: bytes=32 tempo=3ms TTL=64
Resposta de 192.168.0.241: bytes=32 tempo=4ms TTL=64
Resposta de 192.168.0.241: bytes=32 tempo=2ms TTL=64

Estatísticas do Ping para 192.168.0.241:
    Pacotes: Enviados = 4, Recebidos = 4, Perdidos = 0 (0% de
    perda),
Aproximar um número redondo de vezes em milissegundos:
    Mínimo = 1ms, Máximo = 4ms, Média = 2ms
```

Figura II.13: Teste da rede utilizando a função "ping" no prompt do computador.

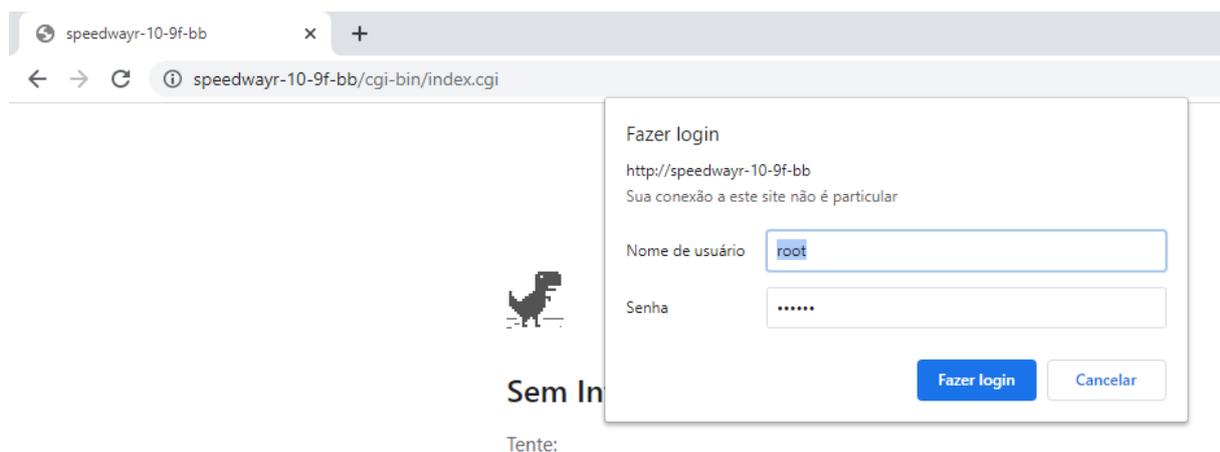


Figura II.14: Usuário e senha para acessar IP do Reader.

Impinj Speedway Revolution x +

Não seguro | speedwayr-10-9f-bb/cgi-bin/index.cgi




READER INFORMATION	
Reader Name	SpeedwayR-10-9F-BB
Uptime	0 Days, 14 hours, 26 minutes, 58 seconds
System Time	Tue Jun 2 09:23:10 UTC 2020
LLRP Status	Disconnected
RFID Status	Idle
STP Status	Not Activated

READER UPGRADE	
Upgrade Status	Ready
Last Operation Status	N/A
Select Upgrade File	<input type="button" value="Escolher arquivo"/> Nenhum arquivo selecionado
Upgrade Now	<input type="button" value="Upgrade"/>

DETAILED INFORMATION	
Model Name	Speedway R420
Regulatory Region	Brazil 902-907 and 915-928 MHz
MAC Address	00:16:25:10:9F:BB
Software Version	4.8.3.240
Hardware Version	250-004-000
Serial Number	370-12-28-0185

CHANGE REGULATORY REGION	
Available Regions	Brazil 902-907 and 915-928 MHz ▾
Press to Update	<input type="button" value="Update"/>

READER REBOOT	
Reboot Status	Ready To Reboot
Press to Reboot	<input type="button" value="Reboot"/>

NETWORK INFORMATION	
IP Address	192.168.0.241
Network Mask	255.255.0.0
Default Route	192.168.0.1
Broadcast Address	192.168.255.255

QUICK LINKS

- www.impinj.com
- support@impinj.com
- [Speedway Installation & Operations Guide](#)



Figura II.15: Interface web do Reader.

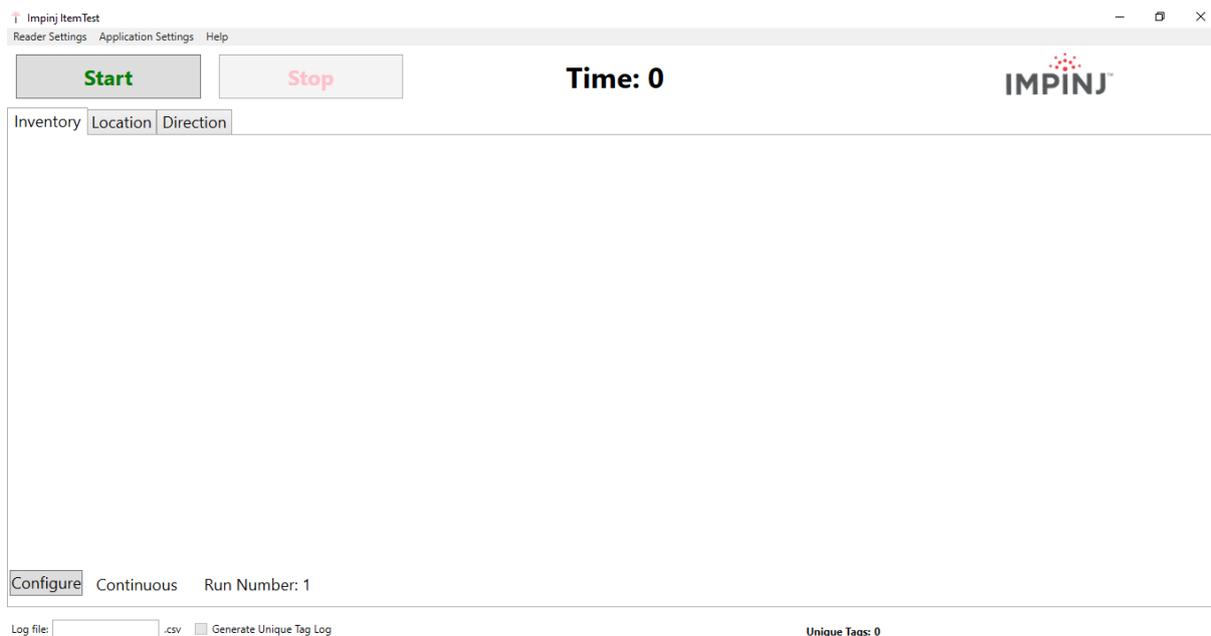


Figura II.16: Tela inicial do aplicativo ItemTest da Impinj.

II.3.3 Teste do sistema

Se tudo ocorreu corretamente até esta etapa, indica que a conexão foi bem sucedida. O próximo passo é testar a funcionalidade do Reader através da leitura de TAGs.

II.3.3.1 Teste utilizado o aplicativo ItemTest

Para isso pode-se verificar rapidamente usando o ItemTest, um aplicativo de teste do Windows da Impinj. Basta baixa-lo no site oficial da Impinj e instalá-lo. A figura II.16 mostra a tela inicial do aplicativo ItemTest.

Após isso, clique em "Reader Settings" para abrir a aba de configurações do Reader, como mostra a figura II.17.

Na página "Reader Settings", no canto inferior esquerdo, clique em "New". No campo "Input Hostname", digite o endereço IP ou o hostname do Reader e depois clique em "OK", como mostra a figura II.18.

Aparecerá uma nova linha na aba do "Reader Settings" com o hostname escolhido. Clique então na caixa "Configure" e configure os campos necessários. Neste exemplo foram conectadas duas antenas ao Reader. É possível também alterar potência e sensibilidade, como é mostrado na figura II.19.

Clique em "OK" e habilite o Reader clicando na caixa "DISABLE" caso ela não esteja habilitada. Dessa forma, se tudo estiver correto você terá uma tela similar a mostrada na figura II.20.

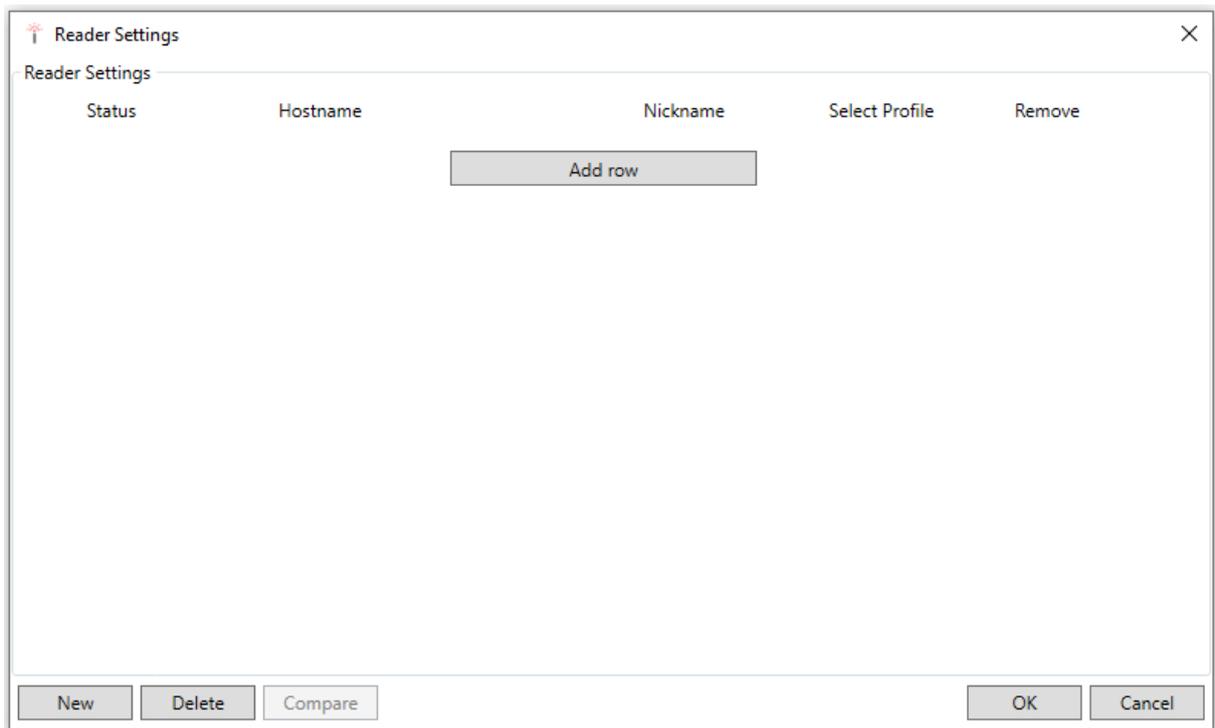


Figura II.17: Tela de configurações do Reader.

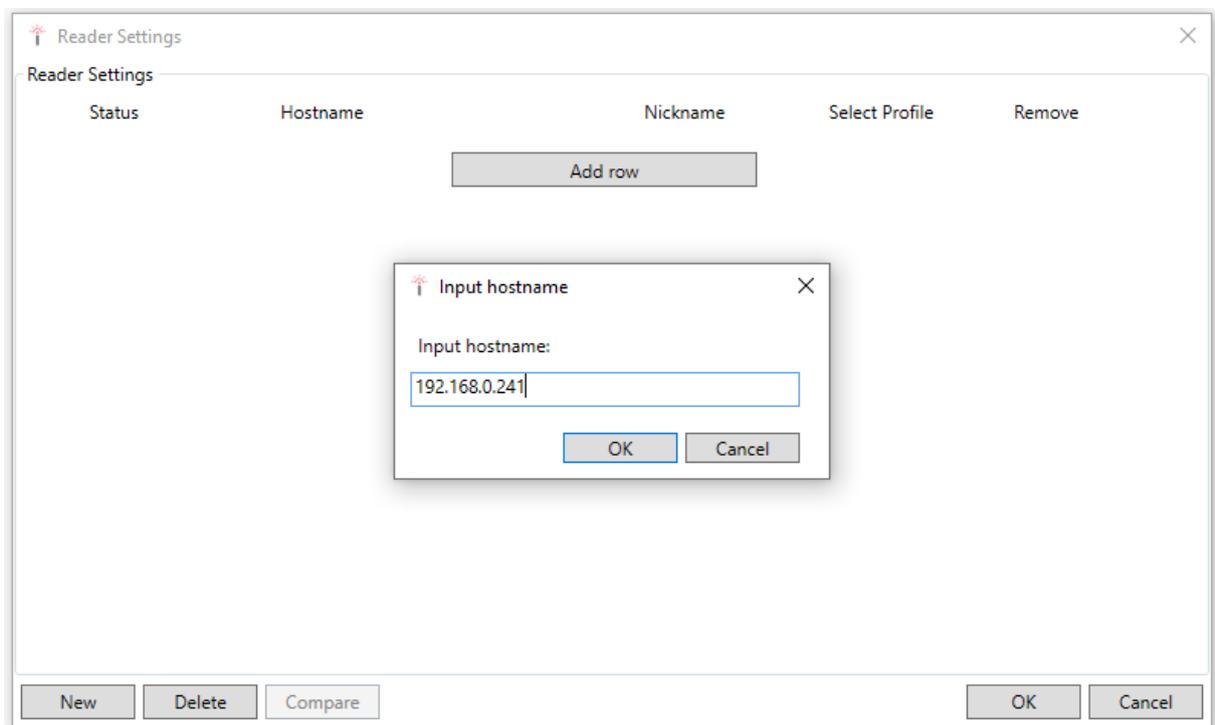


Figura II.18: Configuração de um novo Reader.

Configure 192.168.0.241 Profile: Default Region: Brazil

Inventory Utilities

Manual Settings

Antennas: 2 All Clear

Reader Mode: Mode 1000 : AutoSet Dense Reader

Estimated Population: 32

Session: Session 1

Search Mode: Dual Target

Power and Sensitivity Settings

Power (dBm): 30 ...

Max Rx Sensitivity

Rx Sensitivity: -70

Optional Settings

Filter: FastID

Memory Bank: Epc

Bit Pointer: 32

Length: 16

Low Duty Cycle

Empty Field Timeout (ms): 2000

Field Ping Interval (ms): 250

Save As New Profile OK Cancel

Figura II.19: Configuração da forma de leitura do Reader.

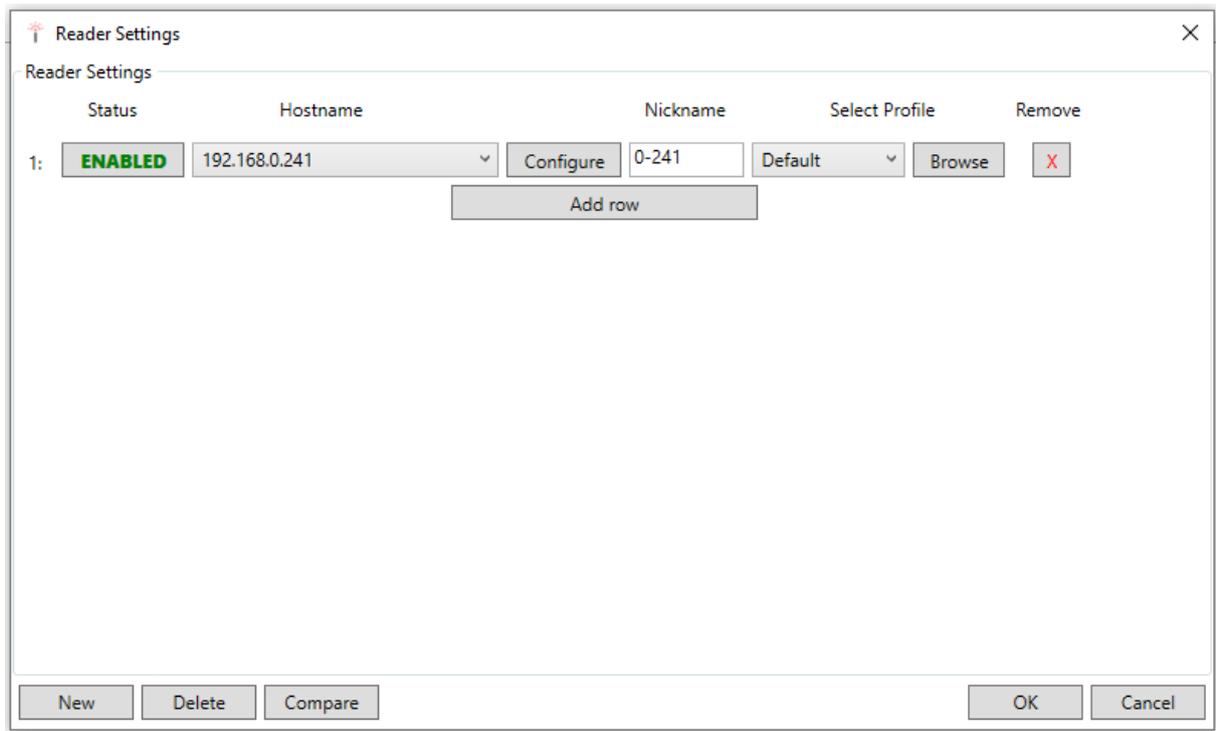


Figura II.20: Reader habilitado na aba de "Reader Settings".

Agora tudo está pronto para ser testado. Clique em "OK" na parte inferior e em "START" na parte superior da tela principal do ItemTest. Posicione as TAGs dentro do campo da zona de leitura do Reader para que elas possam ser lidas, de modo que suas EPCs apareçam. Neste exemplo foram utilizadas 5 TAGs. Como pode-se ver na figura II.21.

The screenshot shows the Impinj ItemTest application window. At the top, there are menu items: Reader Settings, Application Settings, and Help. Below the menu are two buttons: a green 'Start' button and a grey 'Stop' button. In the center, it displays 'Time: 10'. On the right, the IMPINJ logo is visible. Below these elements are tabs for 'Inventory', 'Location', and 'Direction'. The main area contains a table with the following data:

#	EPC	Read # 0-241	
1	E200001B2609014625007765	338	338
2	E200001B2609014624407781	339	339
3	E200001B2609014625807745	339	339
4	E200001B2609014627007715	339	339
5	E200001B26090146277076FD	339	339

Below the table, there is a 'Configure' button, a 'Continuous' checkbox, and 'Run Number: 1'. At the bottom left, there is a 'Log file:' field followed by a '.csv' extension and a 'Generate Unique Tag Log' checkbox. At the bottom right, it shows 'Unique Tags: 5' and 'Rate: 164 tags/s'.

Figura II.21: Leitura das TAGs no ItemTest.

Enquanto as TAGs estão dentro do campo de leitura do Reader, a cor da caixa das TAGs permanecem em um tom claro. Caso se retire as TAGs do campo de leitura, a cor da caixa das TAGs vai se tornando avermelhado, como é mostrado na figura II.22. Dessa forma, sabemos que o Reader está funcionando corretamente.

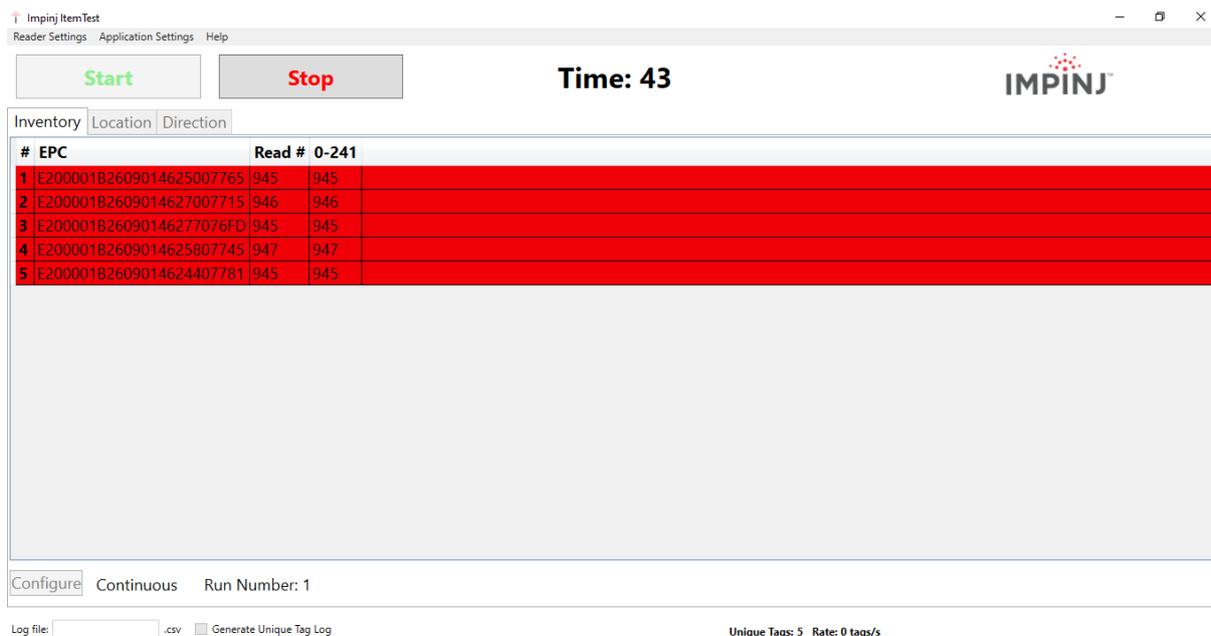


Figura II.22: TAGs fora do campo de leitura do Reader.



Figura II.23: Procura pelo Octane SDK no Visual Studio. Imagem retirada de [6].

II.3.3.2 Teste código exemplo no Visual Studio

Neste trabalho foi utilizado o pacote Octane SDK, da Impinj, para fazer o desenvolvimento da parte de software. O Octane SDK é uma biblioteca de desenvolvimento que suporta aplicativos C sharp e Java para Readers Impinj RAIN RFID. O Octane SDK .NET foi desenvolvido para funcionar no Visual Studio, que é uma IDE da Microsoft, e é possível fazer o download dessa biblioteca diretamente do Visual Studio. Os seguintes passos foram retirados e adaptados do site da Impinj [6].

A primeira etapa é iniciar o Visual Studio e criar um novo projeto. No Gerenciador de Soluções (Solution Explorer), clique com o botão direito do mouse no nome do projeto e selecione "Gerenciar Pacotes NuGet"(Manage NuGet Packages). Na próxima página, clique em "Browse"no canto superior esquerdo, digite "impinj"no campo de pesquisa e pressione a tecla Enter, como mostrado na figura II.23.

Então, aparecerá o conteúdo do pacote Octane SDK como mostrado na figura II.25.

Em seguida, clique na biblioteca que você deseja baixar. O botão "Instalar"aparece localizado na direita. Verifique o conteúdo da pacote, como mostrado na figura II.25.

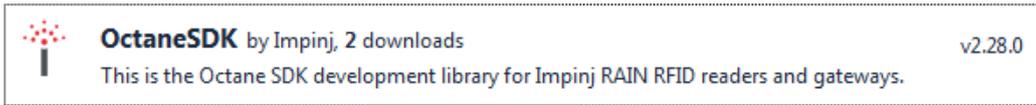


Figura II.24: Conteúdo do pacote no Visual Studio. Imagem retirada de [6].

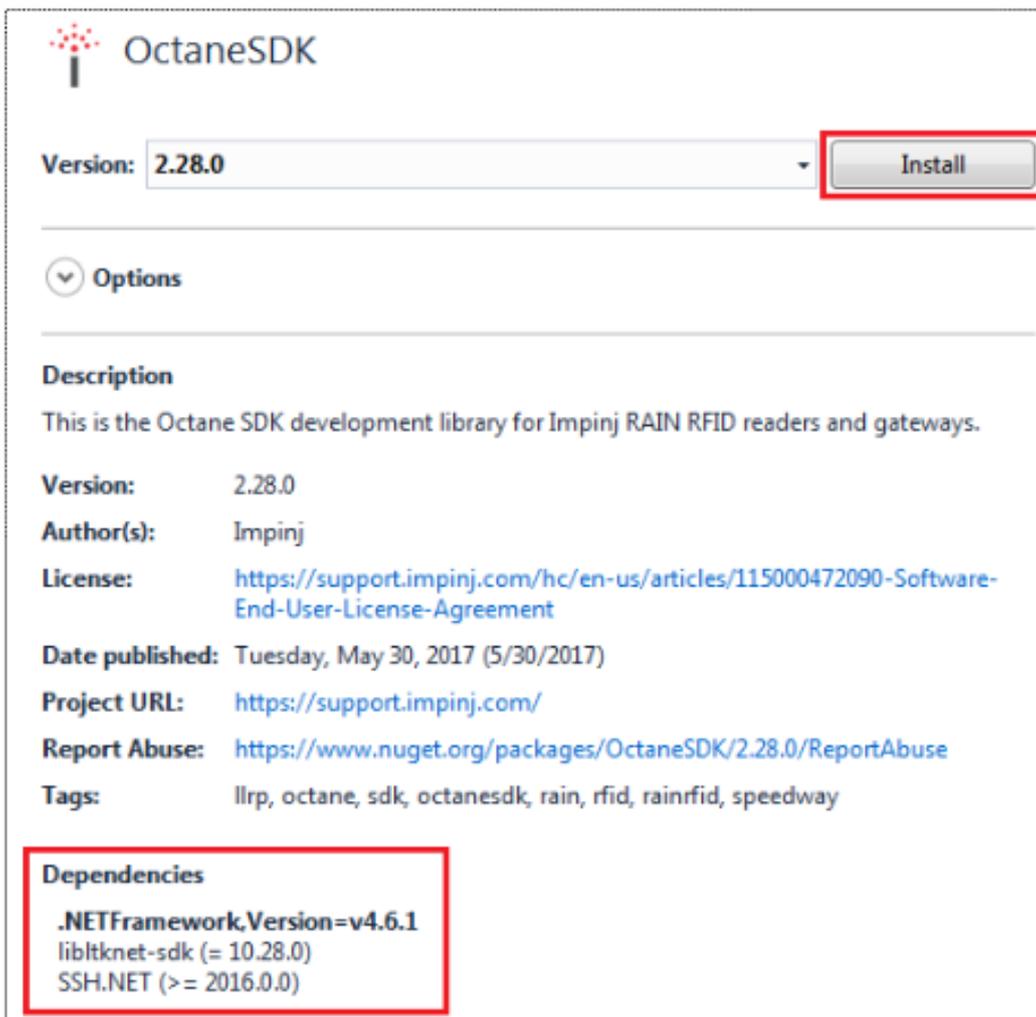


Figura II.25: Botão de instalar do Octane SDK no Visual Studio. Imagem retirada de [6].

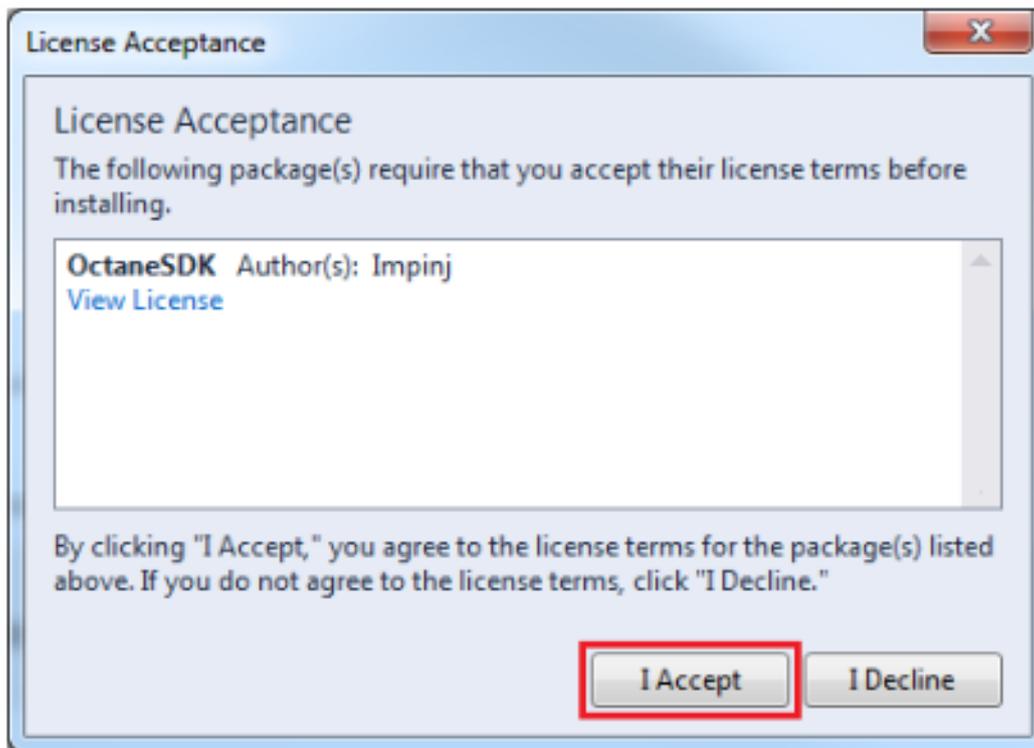


Figura II.26: Aceite os termos da licença para iniciar a instalação no Visual Studio. Imagem retirada de [6].

Verifique as Dependências (no retângulo vermelho acima) antes de instalar o Octane SDK ou o LTK. Clique no botão "Install" para iniciar a instalação e aceite os termos da licença, como mostrado na figura II.26.

Quando a instalação terminar, aparecerá uma mensagem semelhante a mostrada na figura II.27.

Agora que a biblioteca foi instalada, é possível navegar para a pasta de pacotes e carregar o pacote de exemplo no Visual Studio para executá-los. Também é possível fazer o download dos códigos exemplos direto do site oficial da Impinj.

```
Adding package 'OctaneSDK.2.28.0' to folder 'C:\Users\  
Added package 'OctaneSDK.2.28.0' to folder 'C:\Users\  
Added package 'OctaneSDK.2.28.0' to 'packages.config'  
Successfully installed 'OctaneSDK 2.28.0' to SDKNET2.26.1  
Executing nuget actions took 4.2 sec  
Time Elapsed: 00:00:04.6947081  
===== Finished =====
```

Figura II.27: Mensagem indicando que a instalação foi um sucesso. Imagem retirada de [6].

```

7 //
8 // Reader Connectivity example
9 //
10 ///////////////////////////////////////////////////
11
12 using System;
13 using Impinj.OctaneSdk;
14
15 namespace OctaneSdkUseCases
16 {
17     class Program
18     {
19         const string READER_HOSTNAME = "192.168.0.241"; // NEED to set to your speedway!
20         // Create an instance of the ImpinjReader class.
21         static ImpinjReader rReader = new ImpinjReader();
22     }
23 }

```

Figura II.28: Altere o IP do Reader no código do Reader Connectivity.

```

Attempting to connect to My Reader #1 (192.168.0.241).
Successfully connected.
Press enter to exit.
Keepalive received from My Reader #1 (192.168.0.241)
EPC : E200 001B 2609 0146 2580 7745 Timestamp : 1591101160649046
EPC : E200 001B 2609 0146 2440 7781 Timestamp : 1591101160654114
EPC : E200 001B 2609 0146 2500 7765 Timestamp : 1591101160656364
EPC : E200 001B 2609 0146 2700 7715 Timestamp : 1591101160659471
EPC : E200 001B 2609 0146 2770 76FD Timestamp : 1591101160661792
EPC : E200 001B 2609 0146 2500 7765 Timestamp : 1591101160685800
EPC : E200 001B 2609 0146 2580 7745 Timestamp : 1591101160686478
EPC : E200 001B 2609 0146 2700 7715 Timestamp : 1591101160687821
EPC : E200 001B 2609 0146 2440 7781 Timestamp : 1591101160689183
EPC : E200 001B 2609 0146 2770 76FD Timestamp : 1591101160689988
Keepalive received from My Reader #1 (192.168.0.241)
EPC : E200 001B 2609 0146 2770 76FD Timestamp : 1591101165657432
EPC : E200 001B 2609 0146 2440 7781 Timestamp : 1591101165657988
EPC : E200 001B 2609 0146 2580 7745 Timestamp : 1591101165658804
EPC : E200 001B 2609 0146 2700 7715 Timestamp : 1591101165659379
EPC : E200 001B 2609 0146 2500 7765 Timestamp : 1591101165660244
EPC : E200 001B 2609 0146 2500 7765 Timestamp : 1591101165689113
EPC : E200 001B 2609 0146 2440 7781 Timestamp : 1591101165690831
EPC : E200 001B 2609 0146 2580 7745 Timestamp : 1591101165691525
EPC : E200 001B 2609 0146 2770 76FD Timestamp : 1591101165692315
EPC : E200 001B 2609 0146 2700 7715 Timestamp : 1591101165693006
Keepalive received from My Reader #1 (192.168.0.241)
EPC : E200 001B 2609 0146 2700 7715 Timestamp : 1591101170634163
EPC : E200 001B 2609 0146 2500 7765 Timestamp : 1591101170634866
EPC : E200 001B 2609 0146 2770 76FD Timestamp : 1591101170635531

```

Figura II.29: Leitura de TAGs através do código exemplo "ReaderConnectivity"no Visual Studio.

Portanto, para testar se tudo foi instalado corretamente, abra no Visual Studio o código exemplo com o nome de projeto "ReaderConnectivity" e altere no código o IP do Reader utilizado, como mostra a figura II.28.

Para testar, rode o código apertando a tecla "F5", aguarde um pouco a compilação e então coloque TAGs dentro da zona de leitura do Reader. Se a conexão for feita com sucesso e a leitura ocorrer normalmente, uma tela aparecerá com informações semelhantes a mostrada na figura II.29.