

**UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ**

**LUCAS ESDRAS LEGHI GARCIA**

**SISTEMA DE MONITORAMENTO DE ACESSO  
GEORREFERENCIADO PARA INSTITUIÇÕES E EMPRESAS**

**CAMPO MOURÃO**

**2021**

**LUCAS ESDRAS LEGHI GARCIA**

**SISTEMA DE MONITORAMENTO DE ACESSO  
GEORREFERENCIADO PARA INSTITUIÇÕES E EMPRESAS**

**Georeferenced access monitoring system for institutions and companies**

Trabalho de Conclusão de Curso apresentado(a) como requisito para obtenção do título(grau) de Bacharel em Engenharia Eletrônica, do Departamento Acadêmico de Engenharia Eletrônica, da Universidade Tecnológica Federal do Paraná (UTFPR-CM).

Orientador: Prof. Me. Lucas Ricken Garcia

Coorientador: Prof. Dr. Márcio Rodrigues da Cunha

**CAMPO MOURÃO**

**2021**

**LUCAS ESDRAS LEGHI GARCIA**

**SISTEMA DE MONITORAMENTO DE ACESSO GEORREFERENCIADO PARA  
INSTITUIÇÕES E EMPRESAS**

Trabalho de conclusão de curso de Graduação  
apresentado como requisito para obtenção do título  
de Bacharel em Engenharia Eletrônica da  
Universidade Tecnológica Federal do Paraná  
(UTFPR).

Data de aprovação: 23/agosto/2021

---

Eduardo Giometti Bertogna  
Doutorado em Engenharia Elétrica  
Universidade Tecnológica Federal do Paraná

---

André Luiz Regis Monteiro  
Doutorado em Engenharia Elétrica e Informática Industrial  
Universidade Tecnológica Federal do Paraná

---

Lucas Ricken Garcia  
Mestrado em Engenharia Elétrica e Informática Industrial  
Universidade Tecnológica Federal do Paraná

**CAMPO MOURÃO**

**2021**

## AGRADECIMENTOS

Antes de tudo e todos, agradeço a Deus que apesar de todas minhas falhas nunca me fechou as portas, e sempre me deu inúmeras oportunidades em sequência, para que eu possa ter uma vida privilegiada.

Aos meus Pais e irmãos e toda minha família pela paciência e incentivo em todos esses anos de luta, que apesar da minha demora sempre confiaram que um dia este dia chegaria.

A todas as pessoas que já foram meus colegas de classe durante o curso, que sem dúvidas tornaram este processo muito mais fácil trazendo alegria e passando por todos os momentos difíceis juntos incentivando uns aos outros, em destaque ao Ítalo Fernando Rodrigues de Oliveira que já não está mais entre nós.

A todos meus amigos de trabalho da Copel Telecomunicações que assim como meus colegas de classe tornam todo esse processo mais fácil.

Aos amigos Héber Miguel dos Santos e Vinícius de Oliveira Lima que moraram comigo a maior parte do período em que estive em Campo Mourão, se tornando irmãos que podem contar comigo para o resto da vida.

A toda família de Renan Felipe Toshiaki Bieszczad Ikeda e Aline Rocha Leão, que se tornaram minha segunda família enquanto estive em Campo Mourão, em especial a Dona Lídia que sempre me recebeu em sua casa com muito amor e carinho.

Ao meu amigo e Orientador Dr Lucas Ricken Garcia que mesmo distante nos últimos anos sempre esteve ao meu lado quando precisei, me orientando com todo seu conhecimento e paciência.

E por último novamente ao Héber Miguel dos Santos e sua esposa Priscila Galdino dos Santos que cederam seu lar como um local de trabalho com incentivos, para que eu pudesse concluir esse trabalho, já que sempre tive dificuldades de conseguir estudar na minha própria casa.

## RESUMO

O Sistema de Controle de Acesso e Presença Georreferenciado consiste em uma ferramenta baseada na Internet das Coisas (IoT) que tem se mostrado essencial para garantir a segurança e controle da circulação de pessoas em locais onde existe um considerável número de fluxo de indivíduos. Utilizando sistemas de identificação por radio frequência (RFID) e Sistemas de Informação Geográfica (SIG) combinado com linguagem de programação C#, armazenando e buscando as informações em nuvem hospedada na AWS (Amazon Web Services), é possível realizar um mapeamento preciso e intuitivo por onde as pessoas circulam, tanto em empresas como em instituições de ensino de qualquer local conectado a internet. Desta forma foi desenvolvido um sistema capaz de monitorar e controlar a entrada e saída de pessoas, e controlar a lotação de determinados locais utilizando as tecnologias citadas acima.

**Palavras-chave:** Amazon AWS; controle de acesso; internet das coisas; RFID; SIG.

## ABSTRACT

The Geo-referenced Access and Monitoring System for Institutions and Companies consists on a tool based on Internet of Things (IoT) that has proven to be essential to guarantee the security and control of the movement of people in places where there is a considerable number of individuals flow. Using radio frequency identification systems (RFID) and Geographic Information Systems (SIG) combined with C# programming language, storing and searching the information in the cloud hosted on AWS (Amazon Web Services), makes possible to perform a precise and intuitive mapping of where people circulate, both in companies and or educational institutions from any location connected to the internet. Thus, a system was developed capable of monitoring and controlling the entry and exit of people, and control the capacity of certain places using the technologies mentioned above.

**Keywords:** Amazon AWS; access control; internet of things; RFID; GIS.

## LISTA DE ILUSTRAÇÕES

Figura 1 – Exemplo SACS para monitorar crianças em ônibus escolares. . . . .	14
Figura 2 – Exemplo de <i>Shapefile</i> em software de documentação de fibra óptica (FiberWorks). . . . .	16
Figura 3 – Leitor RFID-RC522, TAG tipo chaveiro e cartão <i>RFID</i> . . . . .	19
Figura 4 – Fluxograma do módulo de monitoramento. . . . .	24
Figura 5 – Demonstração do aplicativo, indicando incidências de doenças relacionadas ao mosquito <i>Aedes Aegypti</i> e Hospitais da região. . . . .	24
Figura 6 – Diagrama geral . . . . .	25
Figura 7 – Fluxograma das etapas. . . . .	26
Figura 8 – Periféricos da Raspberry Pi 3. . . . .	27
Figura 9 – <i>GPIO Raspberry pi 3</i> , com <i>SPI</i> em cor Roxa. . . . .	28
Figura 10 – Função que realiza leitura de TAG <i>RFID</i> na <i>Raspberry</i> . . . . .	29
Figura 11 – Módulo de cadastro - programa principal que realiza leitura de TAG <i>RFID</i> na <i>Raspberry</i> . . . . .	30
Figura 12 – Código principal do módulo para controle e registro de entrada de pessoas. . . . .	31
Figura 13 – Código principal do módulo para controle e registro de saída de pessoas. . . . .	32
Figura 14 – Janela de trabalho do <i>MYSQL QUERY BROWSER</i> . . . . .	33
Figura 15 – Figura representando o diagrama ER do banco de dados. . . . .	33
Figura 16 – Representação da função <i>InitDB()</i> . . . . .	35
Figura 17 – Balões indicando nome das ferramentas em vermelho. . . . .	35
Figura 18 – Chamada de escrita e leitura do Banco de dados com a classe <i>DAL</i> . . . . .	36
Figura 19 – Ilustração de um <i>ContextMenuStrip</i> , à esquerda adicionando botões e à direita botões prontos. . . . .	37
Figura 20 – Botões de navegação do mapa. . . . .	37
Figura 21 – Lógica de como Pesquisar Alunos. . . . .	38
Figura 22 – Função utilizada para filtrar log do bloco. . . . .	39
Figura 23 – Função utilizada para pesquisar log por indivíduo. . . . .	40
Figura 24 – Função de consulta ao banco de dados. . . . .	41
Figura 25 – Tela inicial da aplicação desenvolvida. . . . .	42
Figura 26 – Apresentação da funcionalidade de cadastro de usuários. . . . .	43
Figura 27 – Caixa de mensagem para indicar ao usuário o momento de aproximar a TAG. . . . .	43
Figura 28 – Demonstração de falha ao tentar obter TAG no cadastro. . . . .	43
Figura 29 – Retorno na aplicação ao aproximar a TAG e pressionar OK. . . . .	44
Figura 30 – Resultado na interface de desenvolvimento do sistema embarcado. . . . .	44
Figura 31 – Pesquisando usando a aba de cadastros . . . . .	45
Figura 32 – Demonstração do sistema embarcado negando a entrada de TAG não cadastrada. . . . .	45
Figura 33 – Demonstração do Aluno 01 dentro do bloco A . . . . .	46
Figura 34 – Demonstração de como consultar o LOG do bloco A. . . . .	46
Figura 35 – Demonstração da consulta de LOG de cada indivíduo, controle e monitoramento de capacidade. . . . .	47
Fotografia 1 – Um mapa de um relatório feito pelo Dr. John Snow em 1854. . . . .	15
Fotografia 2 – Hardware utilizado por Teixeira (2011). . . . .	23
Quadro 1 – Comparativo de modelos <i>Raspberry PI</i> . . . . .	18
Quadro 2 – Conexão RC522 com <i>Raspberry PI</i> . . . . .	27

## LISTA DE ABREVIATURAS, SIGLAS E ACRÔNIMOS

ANSI	<i>American National Standards Institute</i>
AWS	<i>Amazon Web Services</i> (Serviços Web da Amazon)
ER	Entidade de Relacionamento
GPIO	<i>General Purpose Input/Output</i> (Entrada e saída de propósito geral)
GUI	<i>Graphical User Interface</i>
IaaS	<i>Infrastructure as a Service</i> (Infraestrutura como serviço)
IoT	<i>Internet of Things</i>
IC	<i>Integrated Circuit</i> (Circuito Inter-integrado)
RDS	<i>Relational Database Service</i> (Serviços de Banco de Dados Relacional)
RFID	<i>Radio Frequency Identification</i>
SACS	Sistemas de segurança e controle de acesso
SGBD	Sistema de Gerenciamento de Banco de Dados
SIG	Sistema de informação Geográfica
SMS	<i>Short Message Service</i>
SPI	<i>Serial Peripheral Interface</i> (Interface periférica Serial)
SQL	<i>Structured Query Language</i> (Linguagem de Consulta Estruturada)
SSD	<i>Solid State Drive</i> (Driver de Estado Sólido)
TAG	Rótulo ou etiqueta
TI	Tecnologia da Informação
UART	<i>Universal Asynchronous Receiver/Transmitter</i>
UTFPR	Universidade Tecnológica Federal do Paraná



## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>9</b>
1.1	OBJETIVOS	10
1.1.1	Objetivos gerais	10
1.1.2	Objetivos específicos	10
1.2	JUSTIFICATIVA	10
<b>2</b>	<b>FUNDAMENTAÇÃO TEÓRICA</b>	<b>13</b>
2.1	SACS	13
2.2	SIG	13
2.2.1	<i>ShapeFiles</i>	15
2.3	SISTEMAS EMBARCADOS	16
2.3.1	<i>RASPBERRY PI3</i>	17
2.4	<i>RFID</i>	18
2.4.1	<i>TAG RFID</i>	18
2.5	PERSISTÊNCIA DE DADOS	19
2.6	COMPUTAÇÃO EM NUVEM	20
2.6.1	<i>Amazon.com</i>	21
2.6.2	<i>Amazon Web Services (AWS)</i>	21
2.6.3	<i>Amazon Relational Database Service</i>	21
2.7	CONTROLE <i>ACTIVEX</i>	22
2.8	TRABALHOS CORRELATOS	23
<b>3</b>	<b>METODOLOGIA</b>	<b>25</b>
3.1	MÓDULO MONITORAMENTO DE ACESSO	26
3.2	IMPLEMENTAÇÃO DO BANCO DE DADOS	32
3.3	TESTES DE COMUNICAÇÃO E INTERFACEAMENTO COM O BANCO DE DADOS	34
3.4	APLICAÇÃO EM C# PARA ADMINISTRAÇÃO E VISUALIZAÇÃO DO SISTEMA	34
3.5	TESTES DE INTEGRAÇÃO ENTRE O SISTEMA <i>IN LOCO</i> E APLICAÇÃO EM C#	39
<b>4</b>	<b>RESULTADOS E DISCUSSÕES</b>	<b>42</b>
4.1	CADASTRO	42
4.2	CONTROLE DE ACESSO E VISUALIZAÇÃO	44
4.3	CONTROLE E MONITORAMENTO DE CAPACIDADE	47
<b>5</b>	<b>CONCLUSÕES</b>	<b>49</b>
5.1	TRABALHOS FUTUROS	49
	<b>REFERÊNCIAS</b>	<b>50</b>

## 1 INTRODUÇÃO

Sistemas de segurança e controle de acesso (SACS) têm se tornado uma ferramenta essencial para garantir a segurança de locais onde existe um considerável número de pessoas circulando. Em alguns casos, além de controlar a entrada e saída de pessoas, é possível inserir funcionalidades diversas como a geração de relatórios de fluxo de pessoas e verificar a presença de indivíduos específicos no ambiente (CONTROLE... , 2020). Adicionando-se uma visualização georeferenciada em uma Interface Gráfica do Usuário, do inglês, *Graphical User Interface* (GUI), permite-se que os dados sejam visualizados de forma muito mais amigável e interativa com o usuário final, podendo também ter controle dos locais, escritórios, blocos, salas ou até mesmo vários *campi*, no caso das universidades, em que o indivíduo acessou.

O controle de acesso, por si só, permite aumentar a segurança do espaço físico, não permitindo a entrada de indivíduos estranhos ao local. Quando aplicado a eventos e locais de venda de grande porte (*shoppings* e centros de evento), a possibilidade de gerar relatórios de fluxo de pessoas permite realizar uma análise estatística dos dados, indicando pontos críticos que necessitam de uma maior identificação, avaliação e controle das situações de risco, proporcionando um ambiente mais seguro e agradável para as pessoas que nele circulam.

Essa segurança, obtida por meio do controle de acesso, é essencial principalmente em instituições educacionais. Nestes locais o controle de acesso se torna uma ferramenta primordial para o monitoramento de pessoas. Além de controlar o acesso a locais não permitidos, nestes casos, também é possível auditar presença de alunos e servidores, complementando este processo que, normalmente, é feito de forma manual.

Para maximizar a precisão das informações e facilitar a visualização é possível agregar uma camada com Sistema de Informação Geográfica (SIG) ao *GUI*. Ao adicionar um SIG obtém-se uma referência de estrutura gráfica onde se aproxima a realidade visual do mapeamento dos locais onde se deseja aplicar o controle de acesso.

Juntamente com a necessidade de uma melhoria na segurança, o mundo testemunha o nascimento de uma era onde a Internet pode ser encontrada em qualquer lugar, mostrando-se cada vez mais presente nas atividades cotidianas, conectando indivíduos, eletrodomésticos, veículos e objetos pessoais. A tecnologia que permite esta conexão tem sido chamada de Internet das Coisas, do inglês, *Internet of Things* (IoT), também conhecida como Internet dos Objetos, definida por Vargas (2014) como uma interconexão entre pessoas e objetos, reduzindo possíveis barreiras na

transmissão de informações homem-máquina.

Segundo Noronha *et al.* (2014) a IoT apresenta para organizações públicas e privadas uma oportunidade sem precedentes de automatizar em até 50% processos manuais. Levando em consideração que a grande maioria dos controles de acessos utilizados no Brasil, principalmente em instituições de ensino, ainda são manuais, a IoT pode ser utilizada como uma ferramenta para automatizar, de forma simples e barata os SACS.

## 1.1 OBJETIVOS

### 1.1.1 Objetivos gerais

Desenvolver um SACS com informações de data e horário, identificando usuários com a tecnologia SIG e identificação por rádio frequência, do inglês, *Radio Frequency Identification* (RFID).

### 1.1.2 Objetivos específicos

Este trabalho tem como objetivos específicos:

- Realizar uma revisão de literatura sobre o tema;
- Realizar a leitura RFID utilizando sistema embarcado;
- Realizar a comunicação do sistema embarcado com banco de dados
- Identificar indivíduos que fazem parte de uma organização através da tecnologia RFID;
- Controlar entrada e saída desses indivíduos através de suas identificações RFID;
- Projetar um software gráfico para interface de usuário
- Realizar a comunicação do sistema embarcado com sistema gráfico;

## 1.2 JUSTIFICATIVA

A elaboração deste sistema auxilia na segurança de universidades, empresas entre outros; controlando e identificando a entrada de pessoas, e demonstrando em uma interface

gráfica de usuário para o gerenciador, ou administrador, em números precisos a quantidade de pessoas dentro de determinado local junto com sua capacidade máxima, além de sinalizações *in loco* aos usuários.

Devido à grande circulação de pessoas em instituições e empresas, existe a necessidade de fortalecer o controle dos usuários ao acesso de diferentes ambientes, possibilitando tanto a auditoria através de sistemas internos, quanto o auxílio na segurança interna. A integração de tecnologias SACS e SIG, junto à identificação por RFID, pode ser uma solução interessante para este monitoramento e controle.

Além disso, com o surgimento do coronavírus, torna-se necessário o distanciamento social e uma maior observação na diminuição de pessoas em locais fechados, por isto as secretarias de saúde vêm publicando portarias, como por exemplo a portaria SES Nº 391/2020 do estado de Santa Catarina (SANTA CATARINA, 2020) , com a diminuição do limite de pessoas em determinados locais.

Decretos estão sendo criados para a diminuição da circulação de pessoas. Este fato leva a consequências nas quais um sistema de monitoramento de acesso georreferenciado pode auxiliar. A primeiro é o cumprimento destas leis com um alerta ao número de pessoas permitido e o já existente dentro do local. Outra consequência é que com a diminuição da circulação se pessoas deve-se garantir que indivíduos estranhos não tenham acesso ao local.

Existem inúmeros exemplos do cotidiano em que é possível utilizar esse tipo de sistema para controlar o fluxo de pessoas em ambientes fechados devido a limitação imposta pela pandemia do coronavírus, sendo os exemplos mais comuns comércio em geral, supermercados, universidades, empresas entre outros.

Em se tratando de supermercados o decreto nº 1311/2021 de Maringá/PR exige que limitem em 50% o número máximo de pessoas dentro do estabelecimento (MARINGÁ, 2021). Segundo Gomes (2021) da RFID BRASIL, atualmente, a maioria destes estabelecimentos utilizam um colaborador dedicado a realizar a contagem manual por meio de fichas numeradas que são entregues ao cliente na entrada e recolhidas na saída. Este método de contagem pode ser extremamente impreciso visto que as fichas saem de ordem e se perde o número exato de clientes e colaboradores que se encontram no estabelecimento. A automatização desse processo pode trazer maior precisão e confiabilidade quanto à exatidão da quantidade de indivíduos dentro do mesmo espaço.

Já em locais onde é necessário um controle mais específico, onde cada indivíduo tem

uma identificação distinta e nem todas as pessoas são autorizadas ao acesso, como uma empresa ou universidade, o controle com *RFID* em conjunto de barreiras impeditivas, além de limitar o número de pessoas no mesmo espaço compartilhado, também possibilita o registro da presença dos indivíduos e limitar o acesso de determinadas pessoas a locais específicos, como por exemplo um funcionário da área técnica ter acesso livre ao almoxarifado de uma empresa.

## 2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo apresenta uma revisão da literatura sobre as principais tecnologias e conceitos utilizados neste trabalho, bem como a apresentação de trabalhos correlatos em diferentes áreas de aplicação.

### 2.1 SACS

Segundo Galhardo (2011), SACS "permitem ou restringem o acesso de pessoas ou veículos a determinadas áreas com ou sem limitação de horários, além de registrar o momento do acesso ou tentativa deste". A identificação pode ser feita através de cartões de proximidade, cartões (*TAGs*) *RFID* ou biometria controlados por sistemas embarcados. SACS tem por finalidade proteger o patrimônio físico e pessoas, além de prover registros de eventos que podem ser utilizados para fins de auditoria.

Na Figura 1, pode-se observar um exemplo de SACS desenvolvido para monitorar o acesso de crianças em ônibus escolares. Nota-se, por exemplo, que neste sistema o controle é realizado via *TAGs* cadastradas para cada usuário e leitores são instalados em cada ônibus para sua identificação e controle. Ainda, este sistema é conectado a servidores via internet e possíveis mensagens de aviso podem ser enviadas via *Short Message Service* (SMS). SACS neste tipo de aplicação, contribuem para maior segurança e controle dos pais e responsáveis em relação à locomoção de crianças ao ambiente escolar.

Como citado na introdução, para maximizar a precisão das informações e facilitar a visualização é possível agregar uma camada com SIG ao sistema de administração do SACS. Ao adicionar um SIG obtém-se uma referência de estrutura gráfica onde se aproxima a realidade visual do mapeamento dos locais onde se deseja aplicar o controle de acesso.

### 2.2 SIG

Sistemas de informação geográficas integram uma forma moderna de aquisição de informações, armazenamento, análises e ferramentas de gerenciamento que resolvem problemas relacionados à informação geoespacial. Madeira (2021) conta que este conceito de sistema teve início em 1854 quando uma epidemia se espalhou pelo bairro de Soho em Londres. Neste cenário,

Figura 1 – Exemplo SACS para monitorar crianças em ônibus escolares.



Fonte: Adaptado de Al-Lawati *et al.* (2015, p. 2).

um médico chamado John Snow, atuou na solução da epidemia localizando nos mapas da cidade de Londres os diversos casos de contágio e, assim, relacionando o número de infectados e sua distribuição geográfica.

Ao monitorar o progresso do contágio ao longo do tempo, percebeu-se que algumas áreas estavam relacionadas ao ponto de início da epidemia. Estes dados, portando, instigaram a ideia de georreferenciar uma determinada característica, de modo que fosse possível avaliar o cenário e e auxiliar nas tomadas de decisão. Esta foi uma das primeiras aplicações reais de um SIG (MADEIRA, 2021). A Fotografia 1 demonstra um dos mapas criados pelo Dr. John Snow.

Segundo Chen (2010) as principais ferramentas que um SIG usa são:

- Sensores remotos, que usam uma variedade de dispositivos, como satélites e aeronaves, e métodos como fotografias e radares, para juntar informações de um dado objeto ou área de uma certa distância;
- Ferramentas geográficas, para estudar as características, habitantes, recursos e evoluções do planeta terra e seu ambiente em *Shapefiles*; e
- *Softwares* de visualização, que apresentam informações úteis de dados adquiridos usando imagens 2D e 3D em *Shapefiles* ou animações.

**Fotografia 1 – Um mapa de um relatório feito pelo Dr. John Snow em 1854.**



**Fonte: Snow (1855 apud MAP..., 2021).**

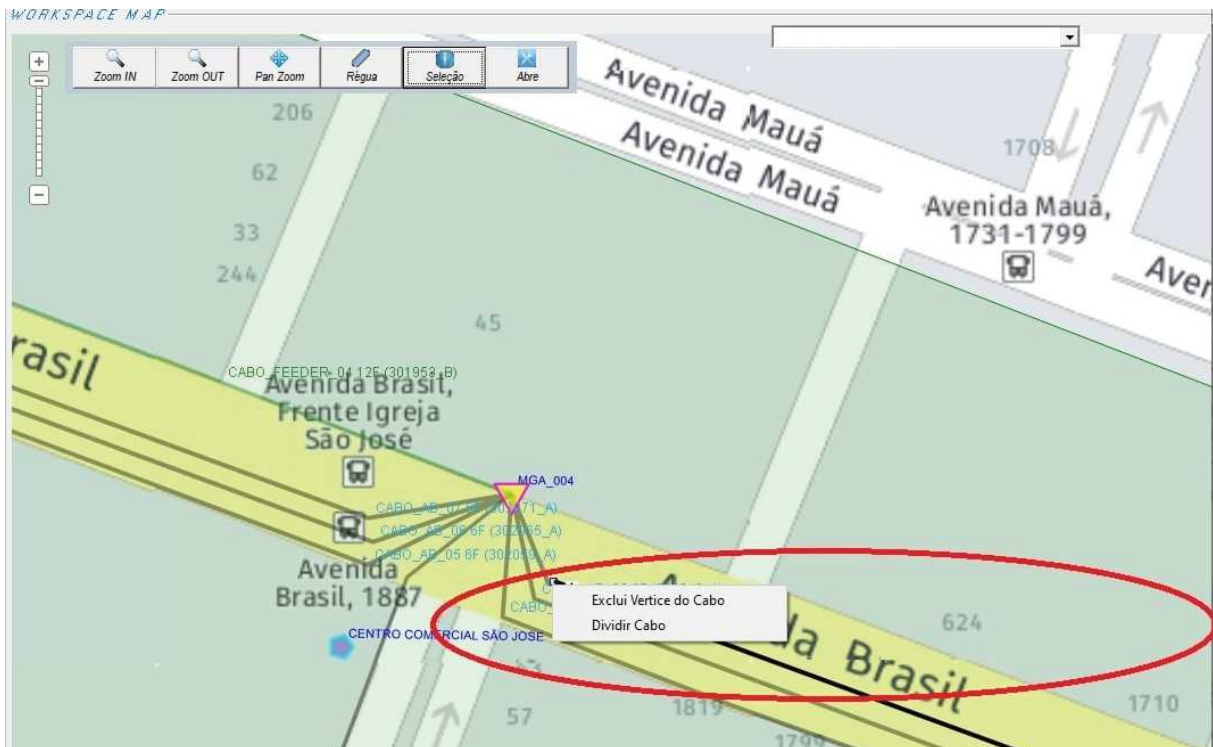
### 2.2.1 ShapeFiles

Segundo Shapefiles (2020), um *shapefile* é um formato vetorial utilizado para o armazenamento de dados, tais como "posição, forma e atributos de feições geográficas". Este armazenamento é realizado em um conjunto de arquivos relacionados e contém uma classe de feição que agrupam feições geográficas semelhantes. "Os *shapefiles* frequentemente contém feições grandes com muitos dados associados e foi historicamente utilizado em aplicativos de *desktop*"(SHAPEFILES, 2020).

A Figura 2 demonstra um exemplo de utilização de *Shapefiles* em um software de documentação de redes de fibra óptica chamado Fiberworks, onde é possível inserir informações de cabos ópticos, caixas de emendas entre outros sobre uma base cartográfica de uma cidade qualquer, além de ser possível interagir com esses mesmos objetos (*ou shapefiles*), inserindo dados de fusões nas caixas de emenda ou visualizando o destino final das fibras contidas dentro de um cabo de fibra óptica (GARCIA, 2021).



**Figura 2 – Exemplo de Shapefile em software de documentação de fibra óptica (FiberWorks).**



Fonte: Garcia (2021)

### 2.3 SISTEMAS EMBARCADOS

Geralmente, tanto SACS quanto SIG podem ser implementados e integrados a sistemas embarcados. Uma vez que, segundo Heath (2002), um sistema embarcado é um sistema microprocessado, que diferentemente de um computador pessoal (comum), é construído para um número limitado de funções específicas com certas restrições. Algumas destas restrições são apresentadas por Stadzisz e Renaux (2008), tais como:

1. Restrições de custos, devido à necessidade de se desenvolver um produto que seja o mais enxuto possível tanto em termos de hardware quanto de software, e se produzir algo barato para atrair o consumidor final.
2. Restrições dimensionais, pois em grande parte dos casos existe a necessidade de que o produto seja prático e ergonômico, ou até mesmo que seja conectado em outro sistema que também possui limitações dimensionais.
3. Restrições de consumo e autonomia, pois existem situações em que o sistema é alimentado por baterias, e tem portanto, uma autonomia limitada de funcionamento.
4. Restrições de recursos de hardware e software, como resultado das restrições anteriores.

Para facilitar o desenvolvimento inicial de sistemas embarcados, têm sido prática comum a utilização de placas de desenvolvimento. Estas, mesmo que apresentando recursos, em grande parte, superiores aos necessários à aplicação, reduzem o tempo de desenvolvimento e podem ser um ponto de partida importante para depois otimizar e desenvolver um hardware específico e que atenda às necessidades do sistema idealizado.

Uma das ferramentas mais conhecidas para o desenvolvimento de sistemas embarcados é a placa de desenvolvimento *Raspberry Pi* que será abordada com mais detalhes a seguir.

### 2.3.1 RASPBERRY PI3

Segundo a Raspberry... (2014), o Raspberry Pi3 é um microcomputador compatível com sistemas operacionais baseados na arquitetura ARMv8-A, sendo que qualquer linguagem compatível com esta arquitetura pode ser compilada neste sistema.

Por possuir um sistema operacional, esta placa de desenvolvimento pode ser usada tanto como um sistema embarcado, como em um microcomputador convencional, já que dependendo do sistema operacional, pode-se utilizar uma interface gráfica, genérica, e periféricos USB (*universal serial bus*) com um baixo custo se comparado a um computador pessoal.

Devido às restrições citadas anteriormente, tratando-se de sistemas embarcados, são utilizados sistemas operacionais com funções reduzidas preocupando-se em obter eficiência e rapidez na execução dos processos destinados à aplicação.

O Quadro 1 apresenta informações técnicas de diferentes modelos da Raspberry Pi. O penúltimo modelo lançado (Raspberry Pi 3) possui grande gama de recursos, dentre estes recursos estão entradas e saídas de propósito geral mais conhecidas como *General Purpose Input/Output* (Entrada e saída de propósito geral) (GPIO), *Serial Peripheral Interface* (Interface periférica Serial) (SPI), *Inter Integrated Circuit* (Circuito Inter-integrado) (I<sup>2</sup>C), *Universal Asynchronous Receiver/Transmitter* (UART), além de possibilitar o uso de periféricos comuns como teclados, mouse, e monitores de vídeo, de maneira prática.

A partir de um sistema como este, baseado no CI BCM 2835, pode-se idealizar um sistema, integrando hardware e software e depois projetar um hardware específico, utilizando apenas dos periféricos necessários. Em contrapartida, dependendo dos recursos, custo de produção e demais fatores, talvez seja de maior interesse permanecer o desenvolvimento na própria placa. Inclusive, versões da Raspberry Pi 3 já são disponibilizadas no Brasil com a devida certificação do sistema sem fio perante a ANATEL (2018).

**Quadro 1 – Comparativo de modelos *Raspberry PI*.**

Modelo	Raspberry PI 3 Modelo B	Raspberry PI zero	Raspberry PI 2 modelo B	Raspberry PI modelo B+
Processador	QuadCortex A53 @ 1,2GHz 64bit	ARM11 @ 1GHz	QuadCortex A7 @ 900MHz	ARM11 @ 700MHz
Set de Instruções	ARMv8-A	ARMv6	ARMv7-A	ARMv6
GPU	400MHz Video Core IV	250MHz Video Core IV	250MHz Video Core IV	250MHz Video Core IV
RAM	1GB SDRAM	512MB SDRAM	1GB SDRAM	512MB SDRAM
Armazenamento	Micro SD	Micro SD	Micro SD	Micro SD
Ethernet	10/100	None	10/100	10/100
Wireless	802.11n/BT 4.0	None	None	None
Saída de Vídeo	HDMI	HDMI	HDMI	HDMI
GPIO	40	40	40	40

Fonte: Adaptado de Benchoff (2016).

## 2.4 *RFID*

Quando se desenvolve sistemas de monitoramento e/ou de controle de acesso, ferramentas de identificação são necessárias para obter os dados do ambiente. Uma tecnologia amplamente utilizada, neste caso, é a *RFID*.

Segundo Violino (2005) *RFID* é um termo genérico usado para descrever um sistema que transmite uma identidade de um objeto ou pessoa usando ondas de rádio. Conforme a What... (2021), um sistema *RFID* pode ser dividido em duas partes: um Rótulo ou etiqueta (*TAG*) e um leitor. As *TAGs RFID* são embarcadas com um transmissor e um receptor. O componente *RFID* nas *TAGs* possuem duas partes: um microchip que armazena e processa as informações e uma antena para receber e transmitir um sinal. Cada *TAG* possui um número de série que pode ser associado a cada objeto ou pessoa específica.

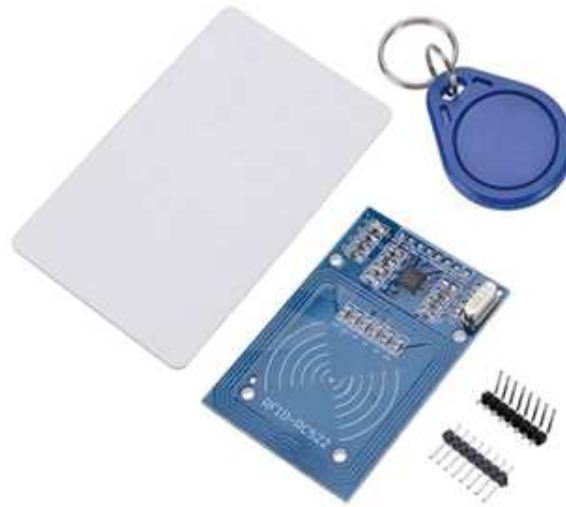
### 2.4.1 *TAG RFID*

Existem dois tipos de *TAG RFID* no mercado: passivas e ativas. A Figura 3 apresenta um exemplo de módulo leitor *RFID* e um receptor passivos, um em forma de chaveiro e outro como um cartão convencional.

Segundo Randall (2017), uma *TAG RFID* passiva armazena dados apenas uma vez em sua fabricação e pode ser lida diversas vezes. Depende muito de sua antena para absorver o sinal de rádio frequência, pois não possui alimentação interna e coleta energia a partir da mesma, esta vem do leitor *RFID*, sendo por estes motivos um tipo de *TAG* de curto alcance.

Segundo Woo-Garcia *et al.* (2016), a captação de energia é possível graças à lei de

**Figura 3 – Leitor RFID-RC522, TAG tipo chaveiro e cartão *RFID*.**



**Fonte: Kit... ([201-?]).**

indução eletromagnética de Faraday. A corrente que flui na bobina do leitor *RFID* produz um campo magnético que ao passar pela bobina da *TAG* induz uma corrente. Esta corrente flui pelo circuito da *TAG* que através de um controlador, altera a carga elétrica sobre sua própria bobina, que gera uma modificação no campo magnético, fazendo com que o leitor perceba esta mudança. Este fenômeno é chamado de modulação de carga, ou do inglês *load modulation*.

Em contrapartida, as *TAGs* ativas se diferem das passivas por possuírem uma fonte de alimentação interna, fazendo com que ela não dependa do leitor para transmitir, possibilitando que a leitura tenha um maior alcance (RANDALL, 2017).

## 2.5 PERSISTÊNCIA DE DADOS

Quando se trata de dados, sejam eles coletados de um questionário, de uma pesquisa ou até mesmo de uma *TAG RFID*, uma das melhores alternativas para se armazenar e manusear os mesmos é a ferramenta chamada banco de dados. Segundo Silberschatz, Sundarshan e Korth (2016, p. 4), "um sistema de banco de dados é uma coleção de dados inter-relacionados e um conjunto de programas que permitem aos usuários acessar e modificar esses dados". Dados que podem representar informações do mundo real e são agrupados de acordo com a suas relações em torno de um mesmo assunto. Como por exemplo, alunos em cursos de engenharia, que podem ser salvos em um grupo chamado "alunos", mas separados dentro deste mesmo grupo de acordo com seu semestre cursado.

Para manipular essas informações são utilizados Sistema de Gerenciamento de Banco de

Dados (SGBD), que além de manipular são capazes de interagir com o programador ou usuário como, por exemplo, o *MySQL* que utiliza a linguagem *Structured Query Language* (Linguagem de Consulta Estruturada) (SQL).

*SQL* é a linguagem de programação padronizada pela *American National Standards Institute* (ANSI) utilizada para se comunicar com banco de dados, criada nos anos 70 por E. F. Codd.

Segundo a *MySQL*. . . (2021), o *MySQL* é o SGBD de código aberto mais conhecido do mundo, que possui "desempenho comprovado, confiabilidade e facilidade de uso". O *MySQL* tornou-se a principal opção de banco de dados, sendo utilizado por varias marcas conhecidas, incluindo *Facebook*, *Twitter* e *Youtube*.

O motivo de tamanha popularidade se deve a sua compatibilidade com os principais sistemas operacionais do mercado, pelo fato de ser desenvolvido em linguagem C e C++, além de que é um programa com código fonte aberto, possibilitando que o usuário faça alterações se necessário. Além disso é possível hospedar esse tipo de banco de dados facilmente em servidores em nuvem de forma simples, sem a necessidade de alocar recursos locais para hospedagem, possibilitando o acesso desses dados de qualquer local do mundo.

## 2.6 COMPUTAÇÃO EM NUVEM

Para Lecheta (2014, p. 24) "computação em nuvem é o termo dado para a possibilidade de acessar seus arquivos como músicas, documentos, vídeos, fotos dentre outros pela internet, usufruindo de recursos infinitos de armazenamento". Neste conceito não existem limites de espaço para armazenamento, a ideia é que se for necessário armazenar informações, independentemente do tamanho, existirá espaço disponível. Porém o termo computação em nuvem pode ser compreendido como algo muito mais amplo do que apenas salvar arquivos na internet. Na verdade, tem significado cada vez o acesso a qualquer recurso de Tecnologia da Informação (TI), sobre demanda pela internet (LECHETA, 2014).

Segundo Mell e Grance (2011) a computação em nuvem tem possibilitado o "acesso a recursos computacionais compartilhados e configuráveis, como redes, servidores, armazenamento, aplicações e serviços que podem ser rapidamente fornecidos e disponibilizados com um mínimo esforço de gestão ou interação com o usuário do serviço".

A computação em nuvem vem sendo cada vez mais adotada por grandes empresas para instalar e disponibilizar as aplicações e os serviços *web*, sendo cada dia mais usado pelos mais

variados desenvolvedores de diversas plataformas de programação. Uma das maiores empresas que oferece esse tipo de serviço é a Amazon.

### 2.6.1 *Amazon.com*

A *Amazon.com* é uma empresa multinacional focada em comércio eletrônico, ou seja, vendas pela internet, e atualmente domina o mercado de varejo online no mundo.

Sua grande diferença em relação às outras lojas, é que a *Amazon* não precisa de um espaço físico para atender seus consumidores, atendendo exclusivamente pelo seu site, se tornando um gigante sistema de *e-commerce*, que é seu grande segredo devido a sua confiabilidade, alta disponibilidade e segurança.

Segundo Lecheta (2014), para chegar a este patamar, a *Amazon.com* fez grandes investimentos em infraestrutura de TI, para expandir seus *datacenters* e atender a toda demanda gerada pela sua loja online, que é acessada por milhões de pessoas ao redor do mundo, trazendo um *know-how* imenso sobre tecnologia, infraestrutura de computadores e sistemas de informação.

### 2.6.2 *Amazon Web Services (AWS)*

Graças a este imenso *know-how* a *Amazon* decidiu compartilhar com o mundo inteiro a sua infraestrutura de servidores possibilitando que qualquer empresa ou desenvolvedor facilmente hospede um site ou serviço nos servidores da *Amazon*.

A partir desta ideia segundo Lecheta (2014) nasce em 2006 uma empresa dentro da *Amazon.com*, denominada *Amazon Web Services* (Serviços Web da Amazon) (AWS), que oferece uma plataforma de serviços completa para desenvolver aplicações em nuvem, em que o usuário inicialmente usufrui dos serviços gratuitamente e conforme aumenta sua demanda pelos serviços paga-se somente o que usar.

### 2.6.3 *Amazon Relational Database Service*

Dentre os principais serviços oferecidos pela AWS esta o *Amazon Relational Database Service* (Serviços de Banco de Dados Relacional) (RDS) que segundo Yamasaki e Aritsugi (2015) é um *Infrastructure as a Service* (Infraestrutura como serviço) (IaaS) que possibilita de maneira simples a utilização de bancos de dados relacionais em nuvem.

Segundo a própria Amazon... (2021) os principais benefícios de seu serviço *RDS* são (i) facilidade para administração, uma vez que torna mais ágil o processo de implantação, sem a necessidade de utilizar infraestrutura própria para instalação e manutenção de banco de dados, (ii) alta escalabilidade, (iii) disponibilidade e durabilidade, (iv) velocidade, uma vez que utiliza armazenamento dados em dispositivos *Solid State Drive* (Driver de Estado Sólido) (SSD) e (v) segurança.

O serviço está disponível em variados modelos de instâncias de banco de dados com desempenho e otimização para memória. Oferece seis diferentes tipos banco de dados: *PostgreSQL*, *Amazon Aurora*, *MySQL*, *Microsoft SQL Server*, *Oracle*, *MariaDB* (AMAZON... , 2021).

## 2.7 CONTROLE *ACTIVEX*

Para Min e Xiaoyan (2007) *ActiveX* é uma tecnologia desenvolvida pela *Microsoft* com base na arquitetura *Component Object Model* (*COM*). Um controle *ActiveX* é um objeto de programa componente que pode ser reutilizado por muitos aplicativos, seja em um único computador ou vários em rede. A tecnologia para a criação de controles *ActiveX* faz parte do conjunto geral de tecnologias *ActiveX*, das quais a principal é o *Component Object Model* (*COM*).

De fato, o controle *ActiveX* não é uma linguagem de programação, mas sim um conjunto de regras a respeito do compartilhamento de informações. É originalmente chamado de '*Object Linking and Embedding*' e usa uma extensão de arquivo *.OCX* (MIN; XIAOYAN, 2007) que pode encapsular pequenos programas, interfaces e funcionalidades específicas. Portanto, permitindo que um programa adicione funcionalidades chamando componentes já desenvolvidos que se integram e aparecem como partes normais do programa.

Ainda segundo Min e Xiaoyan (2007), o usuário do controle *ActiveX* não precisa necessariamente ter conhecimento de como o controle foi desenvolvido, o que facilita o desenvolvimento do software. Sendo assim, este recurso se tornou uma das principais arquiteturas para desenvolver componentes de *software* programáveis (seja em *C*, *C++*, *C*, *Visual Basic* ou *Java*.), desde ferramentas de desenvolvimento de software a ferramentas de produtividade do usuário final. Um exemplo de controle *ActiveX* são SIGs como é o caso do *MapWinGis* e planilhas *Excel* da *Microsoft* que podem ser adicionadas suas funcionalidades dentro de um software escrito em *Visual Basic* ou *C#*.

## 2.8 TRABALHOS CORRELATOS

Na questão sobre monitoramento de acesso, Teixeira (2011) desenvolveu um projeto de funcionalidade bastante similar ao proposto neste trabalho. Neste o autor utilizou *TAGs RFID* passivas junto a um leitor que utiliza frequência de 125 kHz conectado a um microcomputador de mesa ao invés de um sistema embarcado, para implementar um sistema de controle de fluxo de pessoas. O sistema proposto por Teixeira (2011) utilizado pode ser observado na Fotografia 2.

**Fotografia 2 – Hardware utilizado por Teixeira (2011).**



Fonte: Teixeira (2011, p. 52).

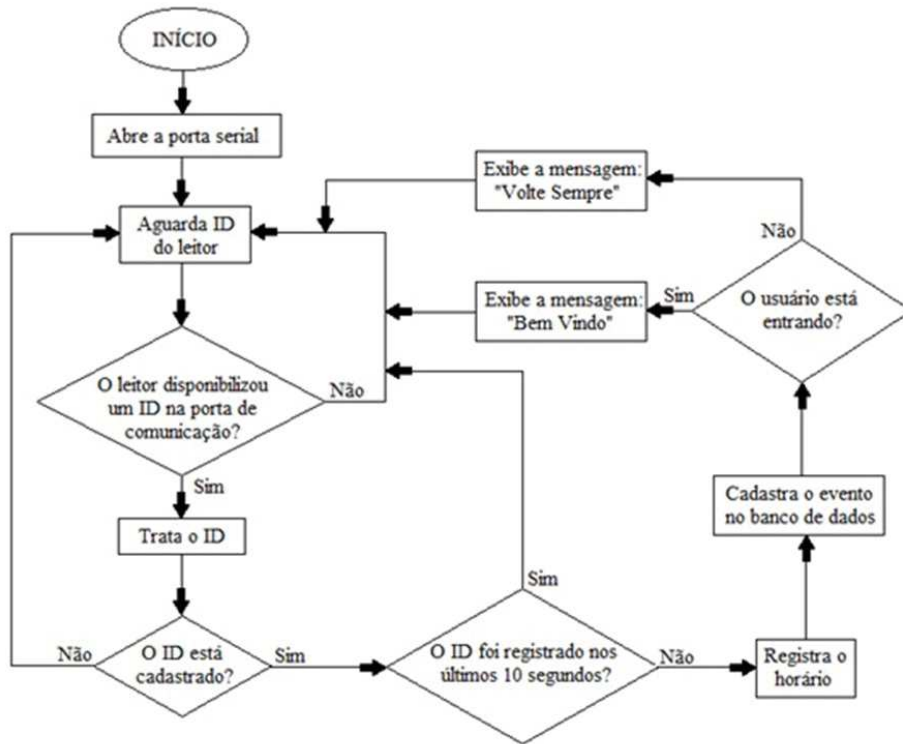
Teixeira (2011) desenvolveu dois softwares utilizando a linguagem de programação Java: O Software de Monitoramento e o Software de Gerenciamento. Estes softwares fazem o registro e a manipulação de informações através de conexões com tabelas de um banco de dados (TEIXEIRA, 2011). A Figura 4 explica qual a lógica utilizada para desenvolver o módulo de monitoramento. Já o módulo de gerenciamento foi utilizado para realizar cadastros e gerar relatórios de entrada e saída dos alunos da sala de aula.

Ao se tratar de SIG, um exemplo a ser comentado é o trabalho feito por Varela (2016) que teve como objetivo a produção de um aplicativo *Android* para rastreamento endêmico das doenças transmitidas pelo mosquito *Aedes Aegypti* no Distrito Federal através da integração com o SIG com apresentação de dados reais disponibilizados pela OMS, OPAS, IBGE e DATASUS, para amostragem gráfica e geográfica, demonstrando estatísticas, gráficos e um mapa com todos os hospitais do Distrito Federal.

A Figura 5 demonstra parte do resultado deste trabalho onde é possível ver pontos vermelhos que representam os hospitais do Distrito Federal e os números representam quantas pessoas em uma determinada região demarcada automaticamente pelo aplicativo, estão com



Figura 4 – Fluxograma do módulo de monitoramento.



Fonte: Teixeira (2011, p. 55).

algum problema de saúde relacionado a doenças transmitidas pelo mosquito Aedes Aegypti.

Figura 5 – Demonstração do aplicativo, indicando incidências de doenças relacionadas ao mosquito Aedes Aegypti e Hospitais da região.

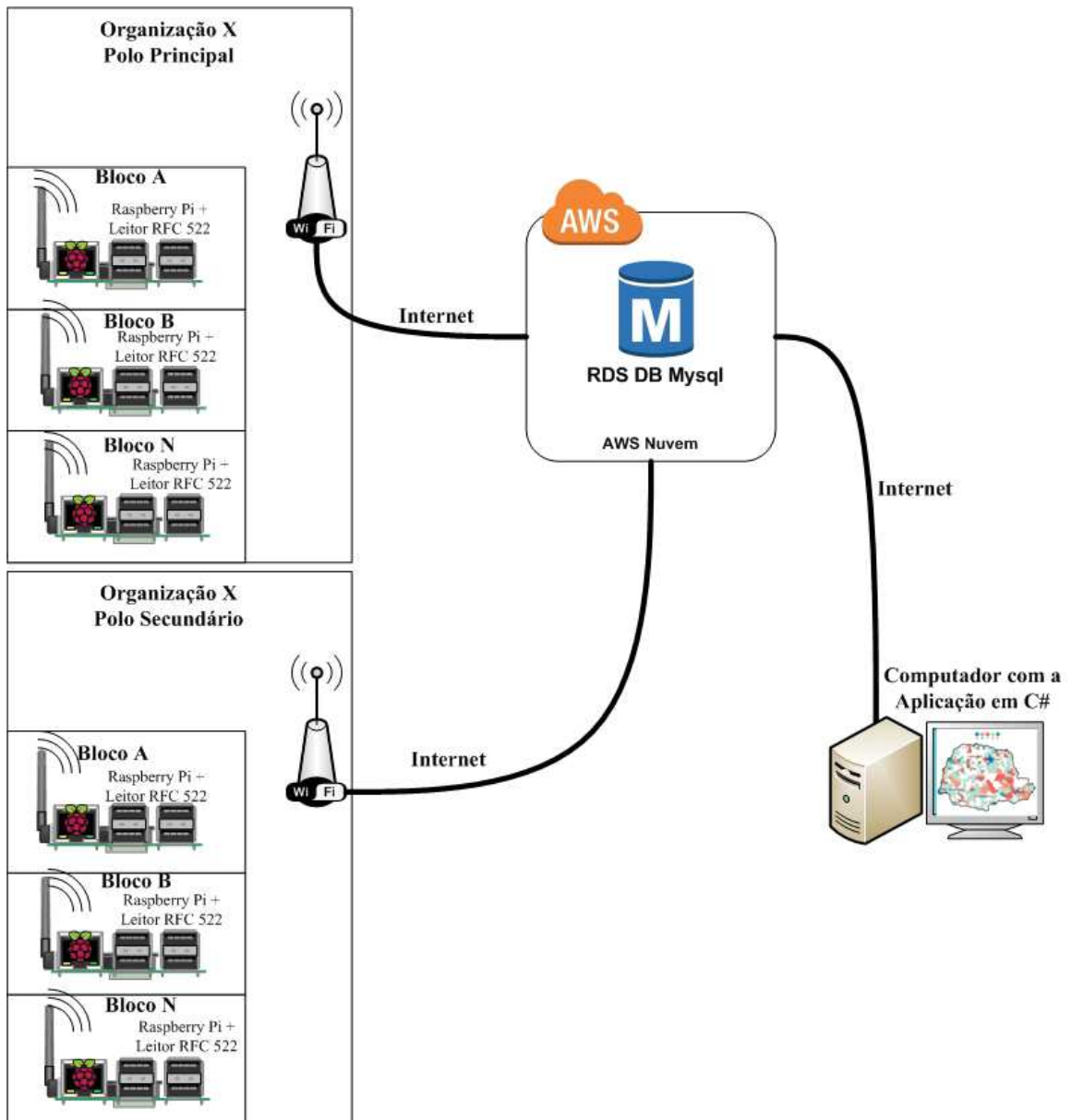


Fonte: Varela (2016, p. 32).

### 3 METODOLOGIA

O presente trabalho foi desenvolvido seguindo o diagrama em blocos apresentado na Figura 6. O sistema de monitoramento de acesso e presença georreferenciado é formado por um *hardware* para monitoramento do acesso *in loco*, *software* para administração do sistema e apresentação visual dos dados, ambos interfaceados por um banco de dados em nuvem. Para facilitar a visualização e compreensão deste trabalho a estrutura física da Universidade Tecnológica Federal do Paraná (UTFPR) de Campo Mourão foi utilizada como exemplo.

Figura 6 – Diagrama geral



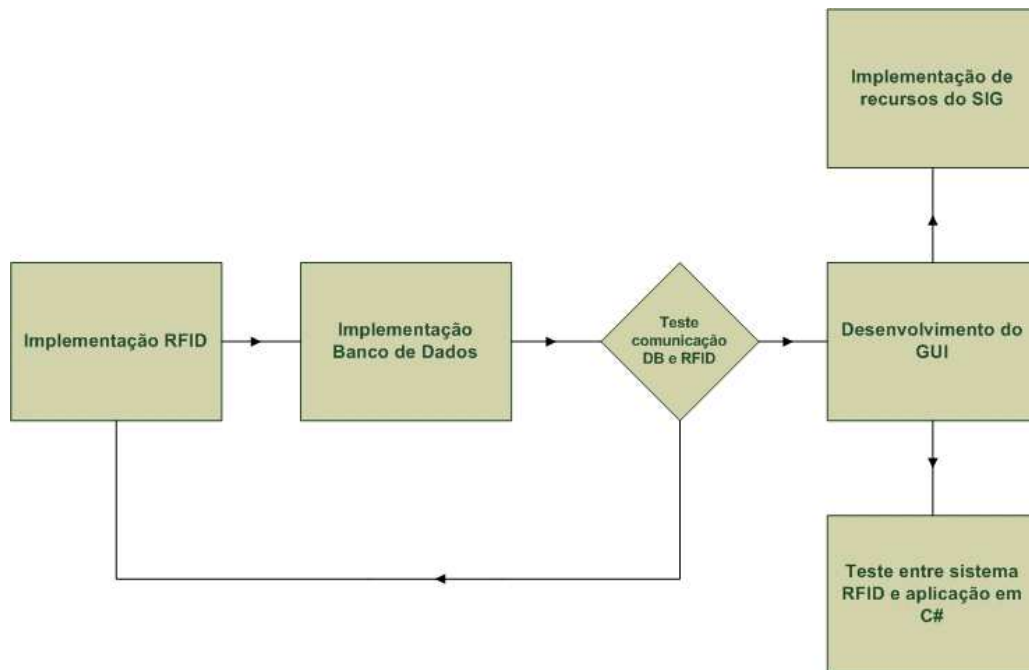
Fonte: Autoria Própria (2021)

O desenvolvimento deste trabalho foi dividido em 5 etapas:

1. implementação do Leitor *RFID* no módulo de monitoramento;
2. implementação do banco de dados;
3. testes de comunicação e interfaceamento com o banco de dados;
4. desenvolvimento do *GUI* e implementação de recursos para o SIG;
5. testes de integração entre o sistema *in loco* e aplicação em C#

A Figura 7 contém um diagrama em blocos das etapas:

**Figura 7 – Fluxograma das etapas.**



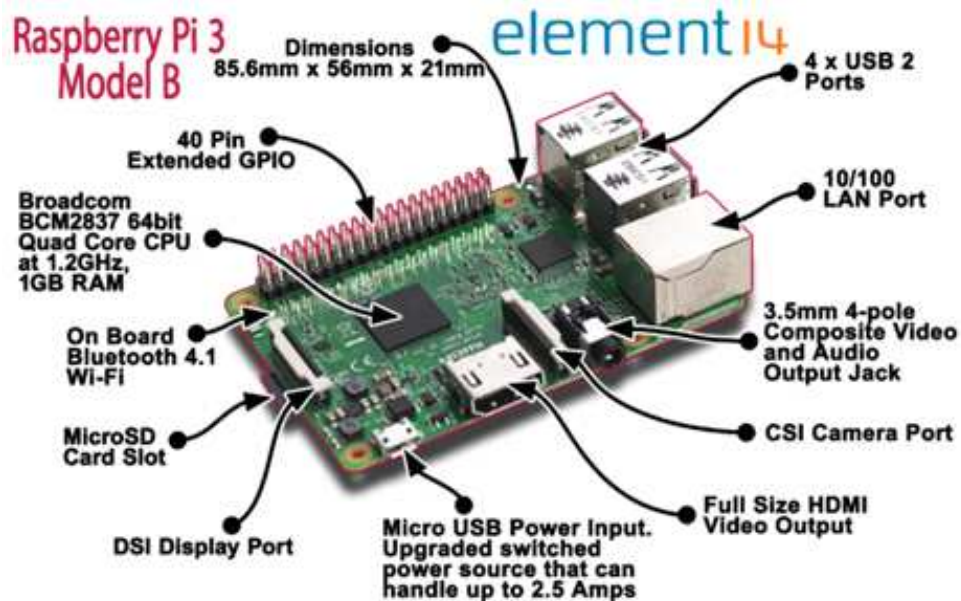
**Fonte: Autoria própria (2021).**

### 3.1 MÓDULO MONITORAMENTO DE ACESSO

Dentre as escolhas principais para realizar este trabalho estavam microcontroladores e sistemas embarcados. Devido a compatibilidade de integração com banco de dados, graças ao sistema operacional que pode ser instalado em um sistema embarcado, o modelo *Raspberry Pi 3* foi escolhido ao invés de um microcontrolador. O uso de um sistema operacional, neste caso, pôde facilitar a implementação do software de leitura *RFID* além de propiciar melhor gerenciamento dos recursos do sistema.

O principal diferencial do modelo Pi 3 em relação aos seus antecessores é a adição de um módulo de comunicação sem fio *wireless* que será de grande utilidade neste trabalho. A Figura 8 apresenta os principais periféricos e conexões disponíveis neste modelo lançado.

Figura 8 – Periféricos da Raspberry Pi 3.



Fonte: Hruska (2019)

Para possibilitar o sistema embarcado de realizar leituras de *TAG RFID* é necessário interfacear o sistema com um Leitor de *RFID* RC522 (Figura 3) através do protocolo *SPI* (*Serial Peripheral interface*) disponível nos pinos em roxo da Figura 9 dos pinos *GPIO* (*General Purpose Input/Output*) da *Raspberry*.

O Quadro 2 demonstra de forma simples e prática como devem ser conectados os pinos do leitor *RFID* RC522 com os pinos *GPIO* da *Raspberry pi 3*, incluindo alimentação.

Quadro 2 – Conexão RC522 com *Raspberry PI*.

Pino RC522	Pino Raspberry PI 3
SDA	24
SCK	23
MOSI	19
MISO	21
IRQ	NC
GND	6
RST	22
3.3v	1

Fonte: Autoria Própria (2021)

Após realizadas todas as conexões foi necessário habilitar o protocolo *SPI* digitando o comando `$raspi - config` nas linhas de comando do sistema operacional *Raspbian* e alterando

**Figura 9 – GPIO Raspberry pi 3, com SPI em cor Roxa.**

**Raspberry Pi 3 GPIO Header**

Pin#	NAME		NAME	Pin#
01	3.3v DC Power		DC Power 5v	02
03	GPIO02 (SDA1 , I <sup>2</sup> C)		DC Power 5v	04
05	GPIO03 (SCL1 , I <sup>2</sup> C)		Ground	06
07	GPIO04 (GPIO_GCLK)		(TXD0) GPIO14	08
09	Ground		(RXD0) GPIO15	10
11	GPIO17 (GPIO_GEN0)		(GPIO_GEN1) GPIO18	12
13	GPIO27 (GPIO_GEN2)		Ground	14
15	GPIO22 (GPIO_GEN3)		(GPIO_GEN4) GPIO23	16
17	3.3v DC Power		(GPIO_GEN5) GPIO24	18
19	GPIO10 (SPI_MOSI)		Ground	20
21	GPIO09 (SPI_MISO)		(GPIO_GEN6) GPIO25	22
23	GPIO11 (SPI_CLK)		(SPI_CE0_N) GPIO08	24
25	Ground		(SPI_CE1_N) GPIO07	26
27	ID_SD (I <sup>2</sup> C ID EEPROM)		(I <sup>2</sup> C ID EEPROM) ID_SC	28
29	GPIO05		Ground	30
31	GPIO06		GPIO12	32
33	GPIO13		Ground	34
35	GPIO19		GPIO16	36
37	GPIO26		GPIO20	38
39	Ground		GPIO21	40

Rev. 2  
29/02/2016

[www.element14.com/RaspberryPi](http://www.element14.com/RaspberryPi)

Fonte: Raspberry... (2016)

para habilitado nas opções de interfaceamento.

Antes de se iniciar o código é necessário instalar a biblioteca do chip de processamento BCM2835 fornecido por McCauley (2021). Após download é necessário instalar o pacote seguindo os seguintes passos abaixo:

---

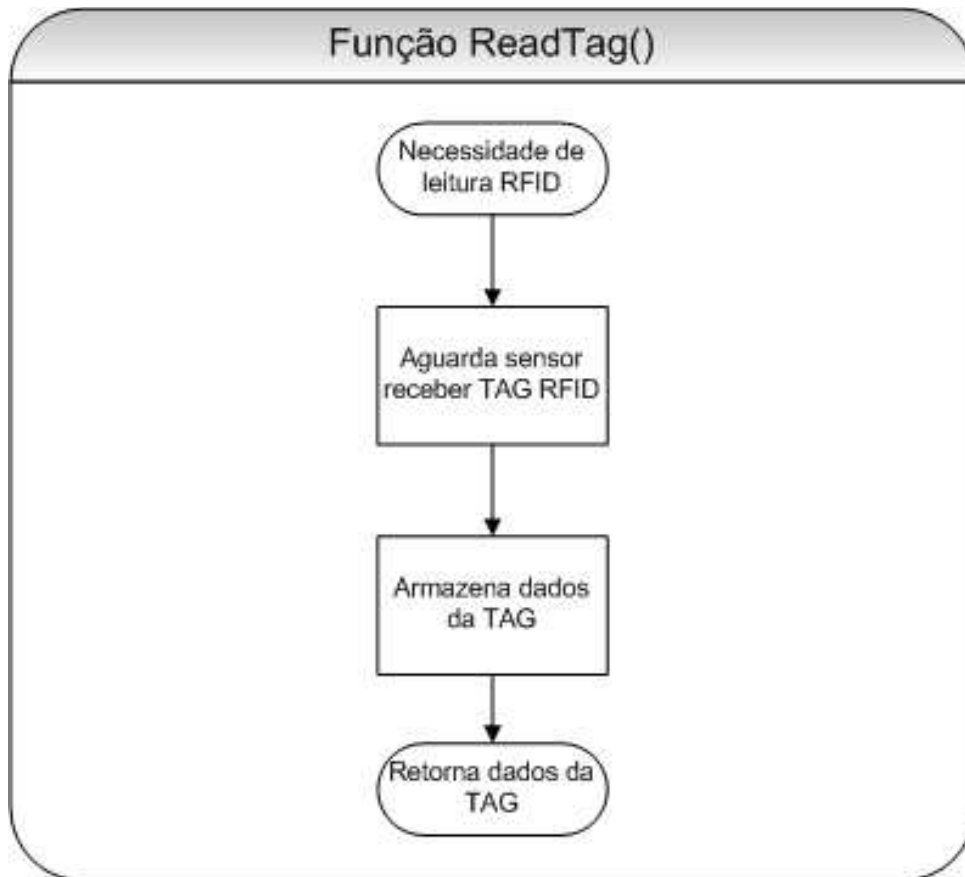
```
tar zxvf bcm2835-1.50.tar.gz
cd bcm2835-1.50
./configure
make
sudo make check
sudo make install
```

---

No desenvolvimento do software para o módulo de monitoramento de acesso, a comunicação com o sensor RC522 teve como base a biblioteca MRFC522 (desenvolvida por Lovbom e Balboa (2018)). A partir dos exemplos de utilização desta biblioteca, realizou-se adaptações para atender as necessidades do projeto, por exemplo, alterando o tipo de retorno das funções para inserção no

banco de dados. A lógica do programa final pode ser compreendida no fluxograma da Figura 10. A função chamada ReadTag() foi utilizada nos programas principais de cadastro de usuários e controle de entrada e saída de usuários.

**Figura 10 – Função que realiza leitura de TAG RFID na Raspberry.**

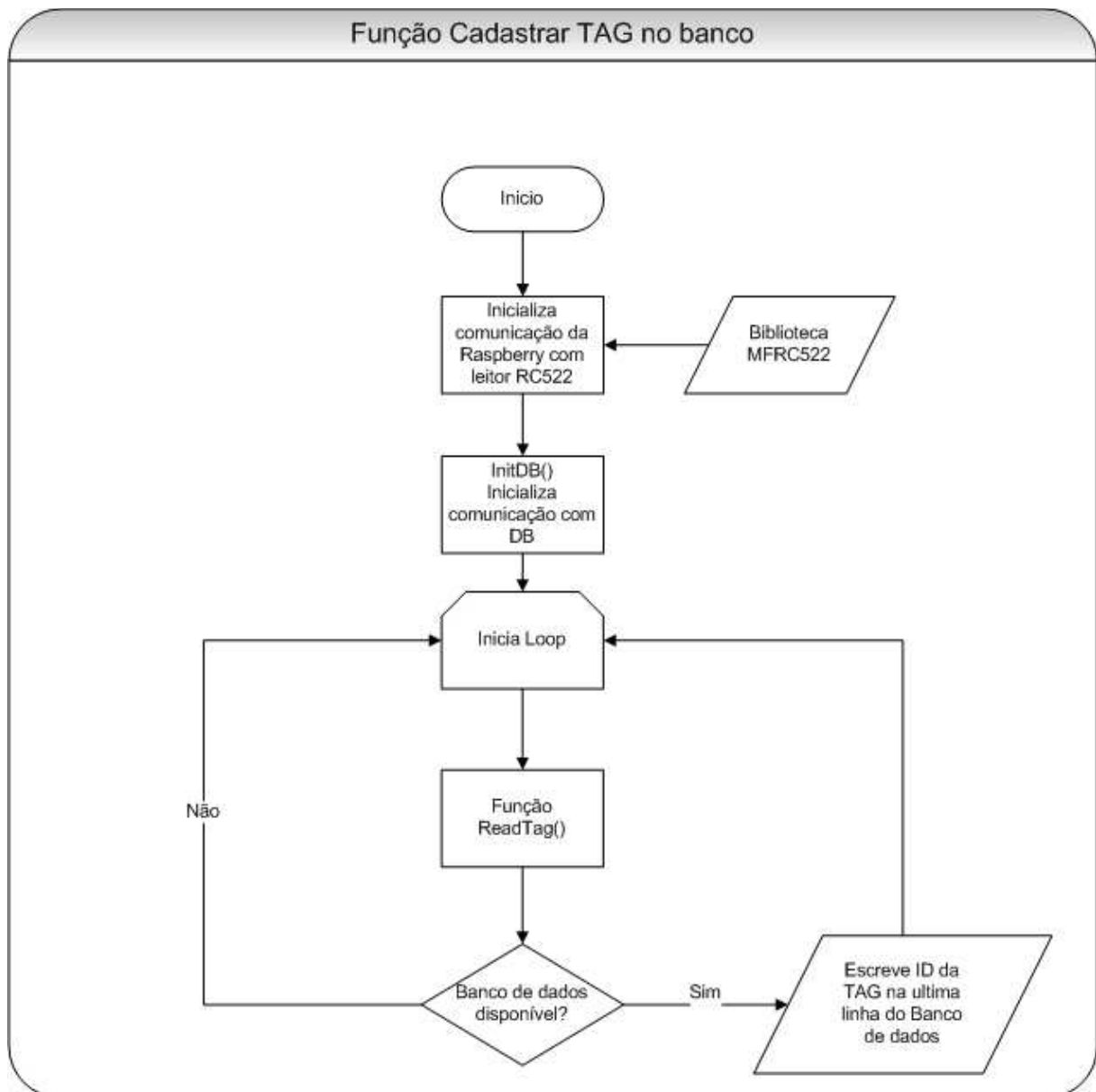


**Fonte: Autoria Própria (2021).**

Para realizar o cadastro de usuários e suas TAGs no banco de dados foi implementada a lógica explicada no fluxograma da Figura 11 em que seu conteúdo é recebido posteriormente pelo GUI para concluir o cadastro. O controle de usuários para entrada e saída dos blocos está representado nos fluxogramas das Figura 12 e Figura 13 respectivamente. Na implementação do protótipo estas ações foram implementadas em programas distintos, sendo executadas via terminal. De fato, para uma aplicação real, ambas ações devem ser integradas em um único programa principal. Cada bloco tem um código com a mesma lógica, porém a escrita é distinta pelo ID de cada bloco. A função InitDB() utilizada para comunicação com o Banco de Dados da Amazon será explicada posteriormente na seção 3.2.

Para compilar o código usou-se o compilador G++, incluso no sistema operacional da Raspberry, usando-se os seguintes comandos:

Figura 11 – Módulo de cadastro - programa principal que realiza leitura de TAG RFID na Raspberry.



Fonte: Autoria Própria (2021).

---

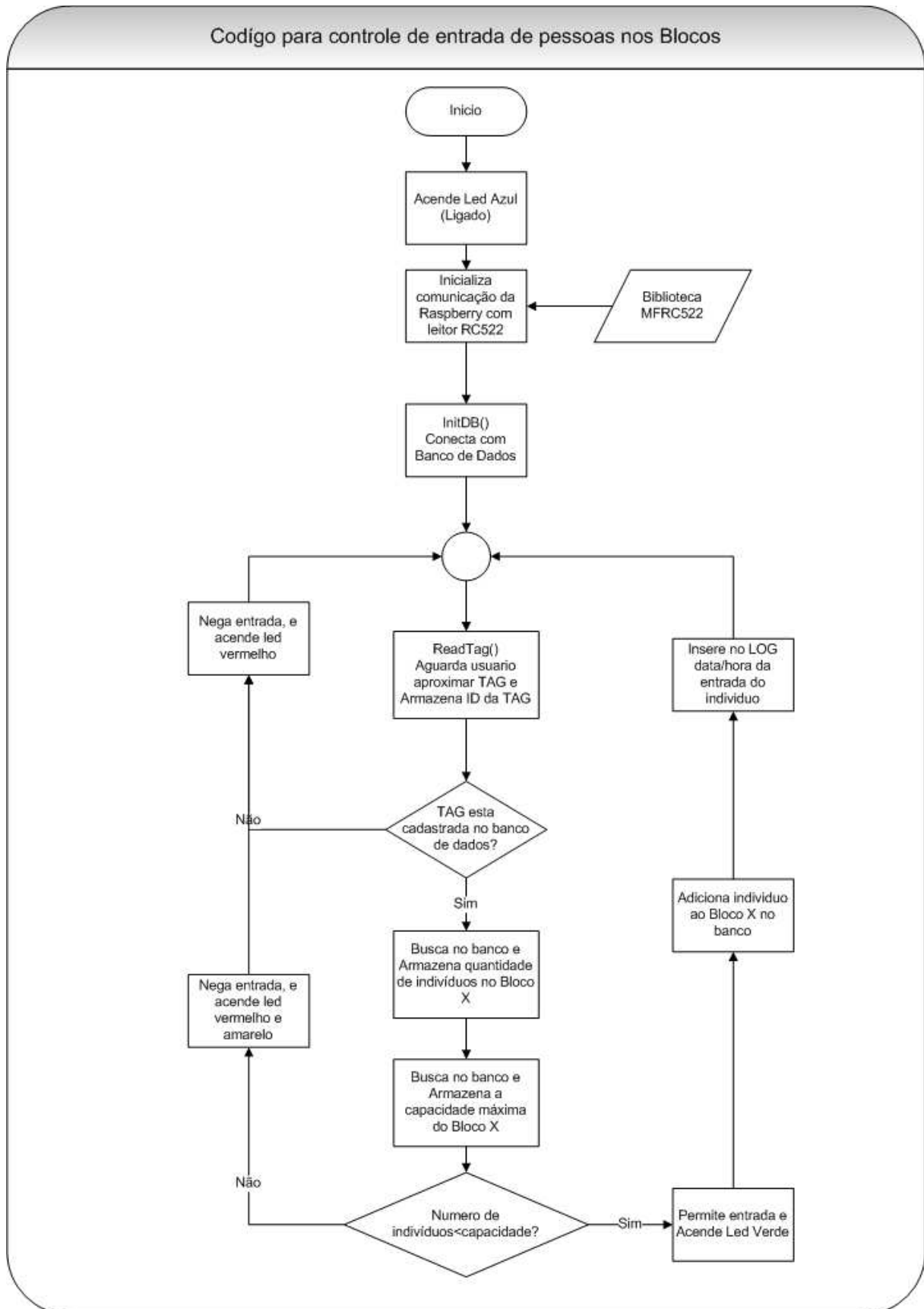
```
g++ MFRC522.cpp "Entrada-Saida-ou-Cadastro".cpp
```

```
-std=c++11 -lbcm2835 -lmariadbclient
```

```
-I/usr/include/mariadb/mysql -O executavel
```

---

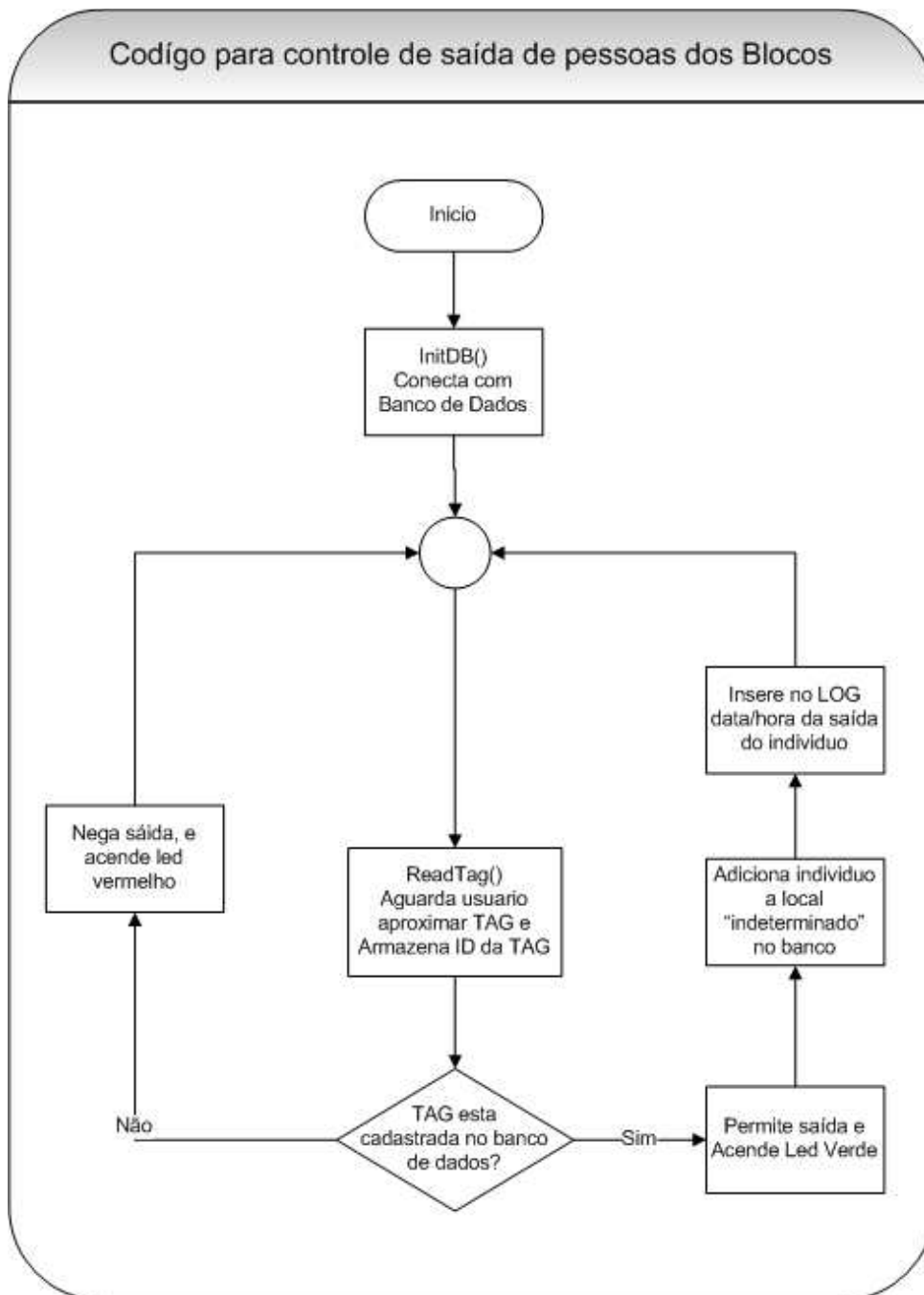
Figura 12 – Código principal do módulo para controle e registro de entrada de pessoas.



Fonte: Autoria Própria (2021).



Figura 13 – Código principal do módulo para controle e registro de saída de pessoas.



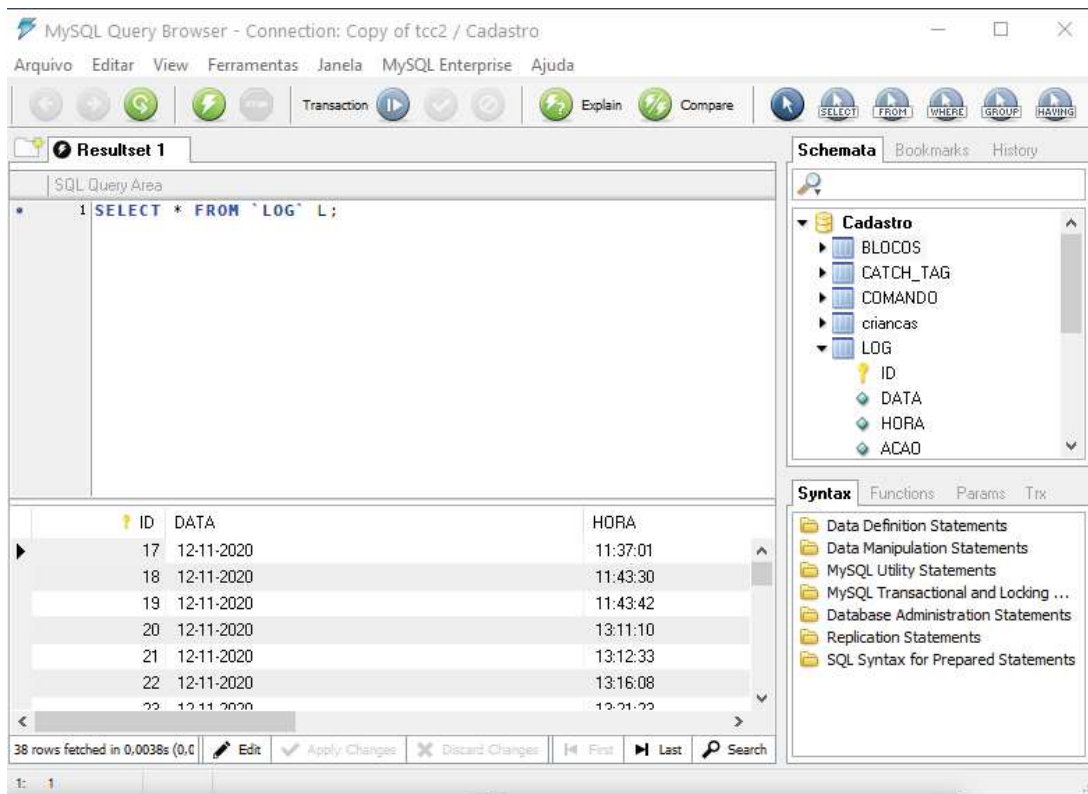
Fonte: Autoria Própria (2021).

### 3.2 IMPLEMENTAÇÃO DO BANCO DE DADOS

De início o Banco de Dados em SQL foi implementado em linha de comando usando o MariaDB do sistema operacional da *Raspberry*, porém conforme o projeto foi evoluindo, e principalmente após hospedar o banco de dados em nuvem, iniciou-se o uso do software *Mysql Query Browser*, que permite o desenvolvimento, consulta e escrita de forma muito mais amigável,

como pode se observar na Figura 14.

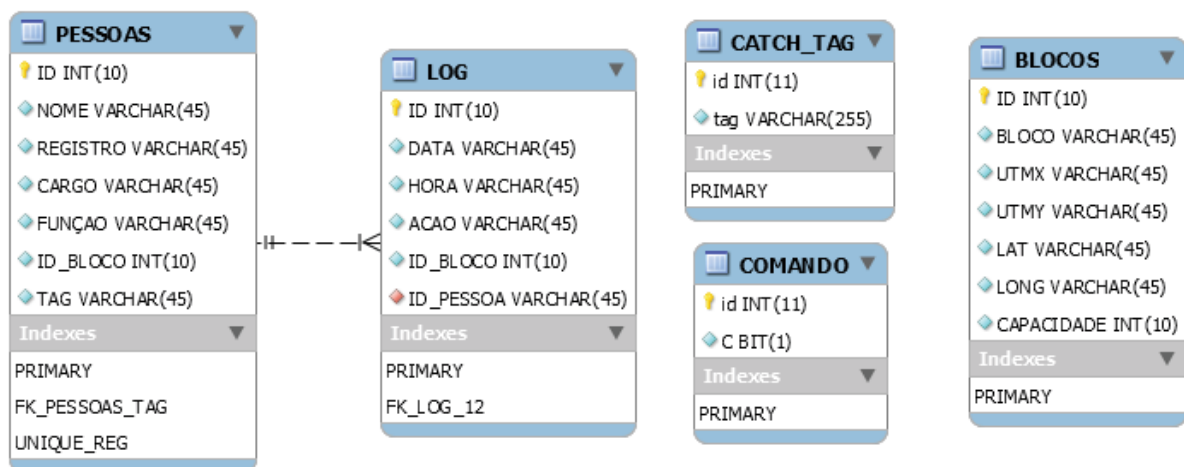
**Figura 14 – Janela de trabalho do *MYSQL QUERY BROWSER*.**



Fonte: Autoria Própria (2021).

As tabelas criadas para funcionalidade do projeto podem ser vistas no digrama de Entidade de Relacionamento (ER) da Figura 15.

**Figura 15 – Figura representando o diagrama ER do banco de dados.**



Fonte: Autoria Própria (2021).

A tabela PESSOAS tem como função armazenar informações sobre cada indivíduo, A coluna ID-BLOCO indica em qual Bloco da universidade o indivíduo se encontra, blocos que são

representados de A até H e Refeitório visualmente e 1 a 9 no banco de dados respectivamente. O numero 10 representa como local indeterminado, caso o indivíduo não se encontre em nenhum bloco.

A tabela BLOCOS tem como função armazenar informações de latitude e longitude e atrelar o número de ID de 1 a 9 de cada bloco da universidade.

A tabela LOG tem como função armazenar o LOG de entrada e saída dos indivíduos, sendo essas informações escritas pelo sistema embarcado e lidas pela aplicação em C#.

A tabela COMANDO envia um bit para o banco de dados que é recebido pelo sistema embarcado como um comando para ler a TAG e escrever o resultado na tabela CATCH-TAG que é lida pela aplicação em C#.

### 3.3 TESTES DE COMUNICAÇÃO E INTERFACEAMENTO COM O BANCO DE DADOS

Os primeiros testes foram realizados com um banco de dados criado localmente no sistema embarcado para testes de escrita e leitura das tabelas. Após isso foram feitos testes de leitura e escrita com um Banco de Dados em rede local, criado em um computador pessoal fora do sistema embarcado para testes do Banco em rede local, e então o banco foi criado no *Amazon RDS* para testes de leitura e escrita por meio da internet.

Sempre que necessária a inicialização do banco de dados o código `initDB.cpp` com a função `dadosDB()` é incluída no código. O código `InitDB` é representado no fluxograma da Figura 16.

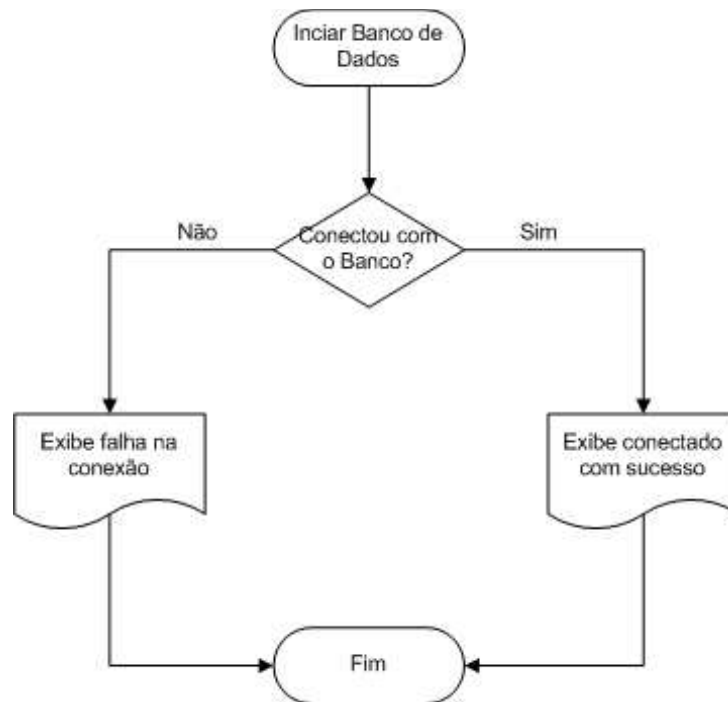
### 3.4 APLICAÇÃO EM C# PARA ADMINISTRAÇÃO E VISUALIZAÇÃO DO SISTEMA

O primeiro passo para se iniciar o desenvolvimento da aplicação em C# é a instalação da ferramenta *Visual Studio 2019*, e criar um novo formulário "*Windows Forms (.NET FRAMEWORK)*". Após isso é necessário instalar o pacote `MYSQL` e adicionar ao visual studio a referência `Mysql.data.dll` no gerenciador de referências.

O segundo passo é instalar o controle `ActiveX` do `Mapwingis` no computador e adicionar o componente `OCX` do `Mapwingis` também no gerenciador de referências. A ferramenta *MapControl* deverá aparecer na caixa de ferramentas.

Tendo sido realizado a adição de referências agora é possível montar o layout do *GUI* usando as caixas ferramentas do *Visual Studio*, como por exemplo *Button* para adicionar botões

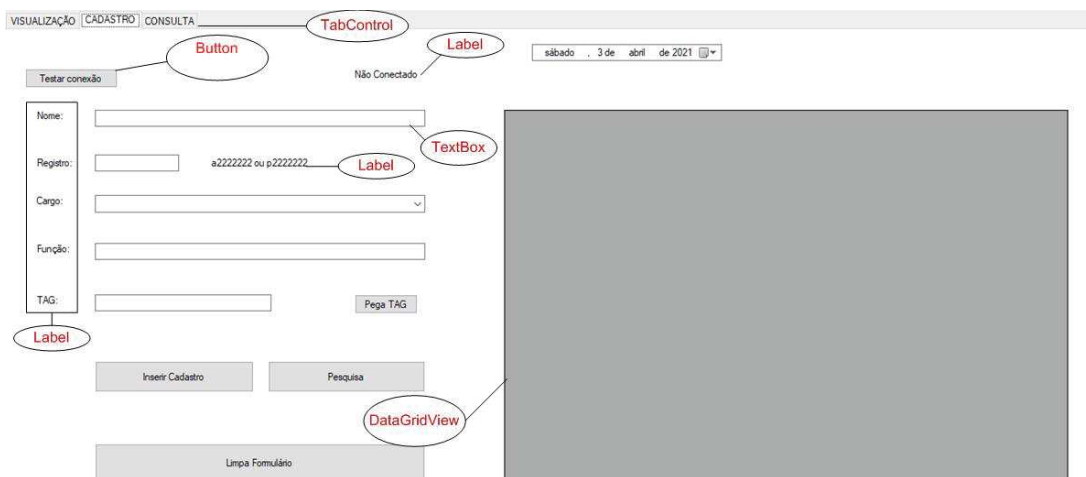
**Figura 16 – Representação da função InitDB().**



Fonte: Autoria Própria (2021).

que geram eventos ao serem clicados pelo usuário, *TabControl*, para gerar guias, *TextBox* permite que o usuário insira textos ou o software exiba textos e *DataGridView* tem como função exibir dados coletados do Banco de Dados. A Figura 17 ilustra exemplos dessas ferramentas.

**Figura 17 – Balões indicando nome das ferramentas em vermelho.**



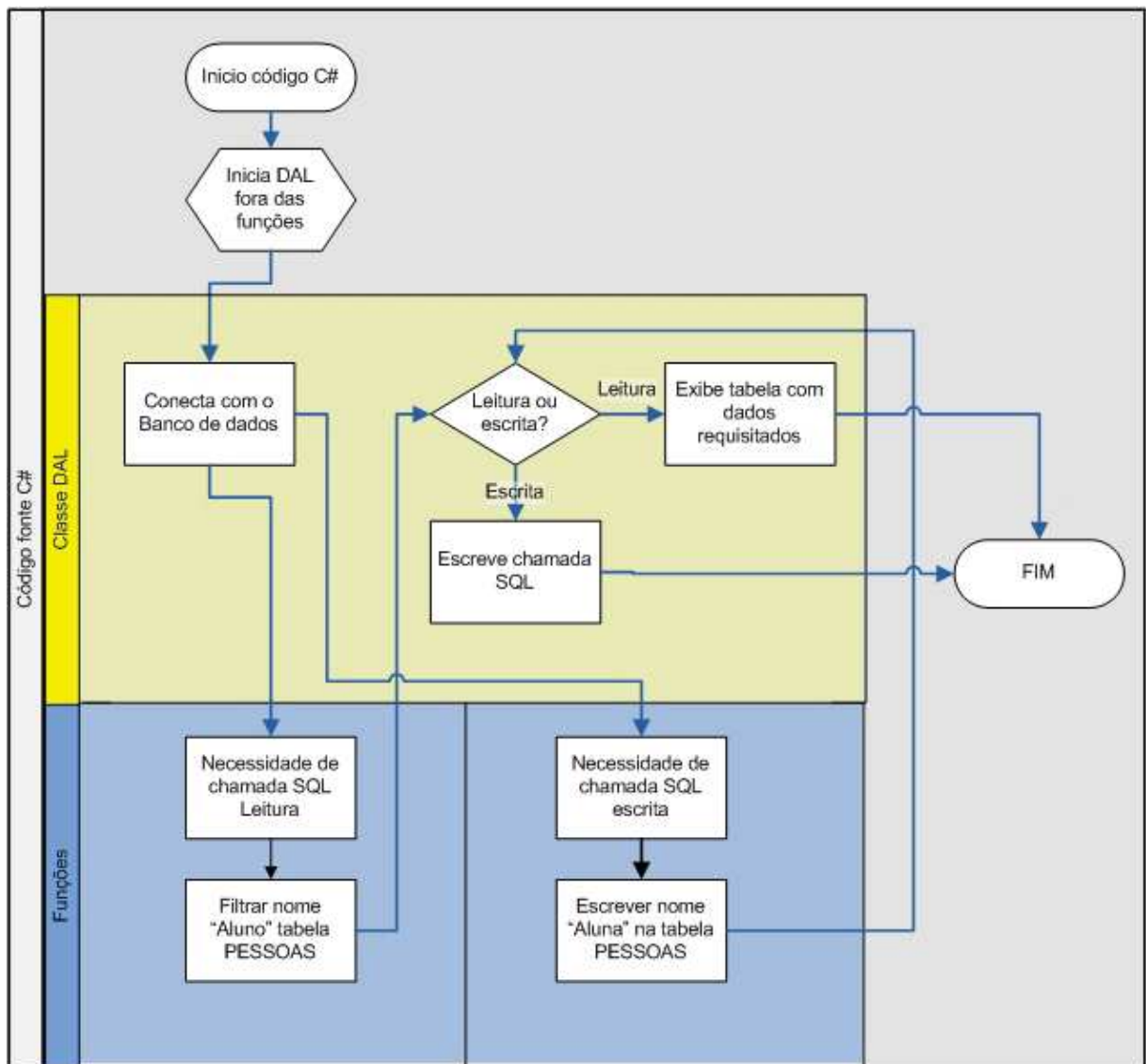
Fonte: Autoria Própria (2021).

É possível alterar as propriedades de cada ferramenta criada em seu *Designer*, como por exemplo, o texto de um *Label*, na aba "Propriedades" do *Visual Studio*. Para programar eventos de uma ferramenta como o *Button*, basta dar dois cliques para ser redirecionado automaticamente

para a parte do código onde é possível manipular sua função escrevendo o código na linguagem de programação C#.

Para a maior parte das chamadas SQL, seja de escrita ou leitura, foi criada uma classe chamada DAL que já contém as informações de autenticação para acesso ao banco hospedado na AWS que com uma simples chamada pode gerar uma tabela (*DataTable*) contendo os dados solicitados ou escrever informações no banco como no fluxograma da Figura 18 , onde é feita a requisição de todas as pessoas com nome de Aluno na tabela PESSOAS e automaticamente retorna uma tabela na tela, ou escreve o nome Aluna na mesma tabela.

**Figura 18 – Chamada de escrita e leitura do Banco de dados com a classe DAL.**



Fonte: Autoria Própria (2021).

Para representar objetos ou locais no mapa foram criados os chamados *ShapeFiles* no código em C#. Para cada bloco da universidade foi criada um *shapefile*. O código de criação dos

*shapefiles* de todos os blocos pode ser encontrado em Leghi-Garcia (2021) nas linhas 136 até 429.

Após isso, tem-se os blocos em suas respectivas coordenadas desenhados sobre o mapa quando o aplicativo é executado. Para interagir com os blocos criados pode-se adicionar um menu que é acionado ao clicar com o botão direito do mouse chamado *ContextMenuStrip* como na Figura 19.

Figura 19 – Ilustração de um *ContextMenuStrip*, à esquerda adicionando botões e à direita botões prontos.



Fonte: Autoria Própria (2021).

Para manusear o mapa deve-se adicionar botões como na Figura 20 e adicionam-se suas funções após dois cliques nos mesmos, redirecionando automaticamente para editar seu respectivo código, como se pode ver no código para o botão para aproximar o zoom:

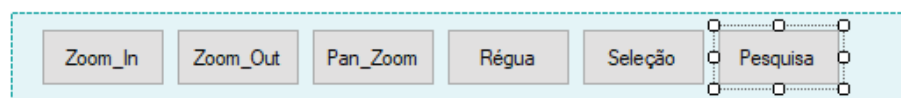
---

```
private void zoom_in_Click(object sender, EventArgs e)
{
    axMap1.CursorMode = tkCursorMode.cmZoomIn;
    //comando de aproximar zoom da biblioteca MapWinGis
}

```

---

Figura 20 – Botões de navegação do mapa.



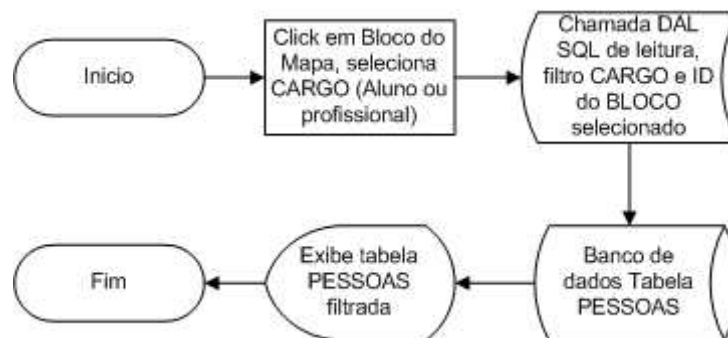
Fonte: Autoria Própria (2021).

Para que o programa identifique qual o objeto clicado no mapa adiciona-se ao programa a função abaixo, que adiciona ao inteiro *id-layer* a identificação do *shapefile* que foi selecionado.

Por conveniência adiciona-se 1 pois os shapes começam em 0. Caso o resultado final seja 1, ao buscar na tabela pessoas do banco de dados a variável ID-BLOCO, este número representará o Bloco A, se retornar o número 2, representará o bloco B e assim por diante. As linhas 686, 687 e 688 do Leghi-Garcia (2021) servem para abrir o MenuContextStrip quando o objeto recebe o clique.

Voltando para o MenuContextStrip, com a variável id-layer pode-se realizar buscas de pessoas que no momento do clique estão dentro dos blocos selecionados, separando por alunos ou professores e servidores. O fluxograma da Figura 21 apresenta a lógica de funcionamento para buscas de alunos dentro do bloco selecionado:

**Figura 21 – Lógica de como Pesquisar Alunos.**



**Fonte: Autoria Própria (2021).**

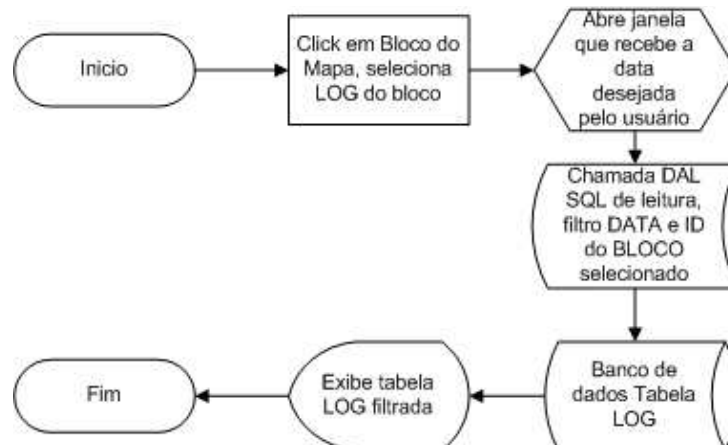
Os registros de entrada e saída (*LOGS*) estão separados por blocos, e é necessário escolher uma data com dia, mês e ano. Para isso foi criada uma janela de recebimento de dados digitados, chamado de Prompt. Após selecionar a opção "Log do bloco" no MenuContextStrip, aparecerá a janela de *prompt* para escolher a data, para isso foi necessário incluir o *prompt* no código do botão de Log do Bloco como na Figura 22.

Após o recebimento da data selecionada, a tabela LOG é filtrada por data e bloco, onde o bloco foi retirado da variável id-layer no momento em que o bloco foi selecionado com o clique no mapa.

Para filtrar o log por indivíduo é necessário ir a aba consulta e digitar ou o nome ou o registro da pessoa. Esta pesquisa tem um funcionamento um pouco diferente. Primeiramente a busca é realizada pelo registro da pessoa na tabela 'PESSOAS' do banco de dados. Identificando o indivíduo na tabela, é retornado a identificação de sua TAG do resultado. Após o resultado é feita a busca na tabela LOG a partir de sua TAG, como pode ser observado no fluxograma da Figura 23.

A mesma lógica é aplicada para a busca por nome, a primeira busca ao invés de filtrar

**Figura 22 – Função utilizada para filtrar log do bloco.**



**Fonte: Autoria Própria (2021).**

por registro, filtra por nome. A função acima é chamada na função que detecta mudança na caixa de texto.

Na aba cadastro, Figura 17 também é possível realizar consultas, a partir do momento que qualquer um dos campos é alterado a pesquisa já é feita utilizando a função `consultaBanco()` representada na Figura 24:

A inserção de dados de cadastro é feita pelo botão "Inserir Cadastro", Figura 17, que contém o código uma lógica similar a Figura 24, porém ao invés de requisitar uma leitura ela efetua uma escrita no banco de dados na tabela 'PESSOAS'.

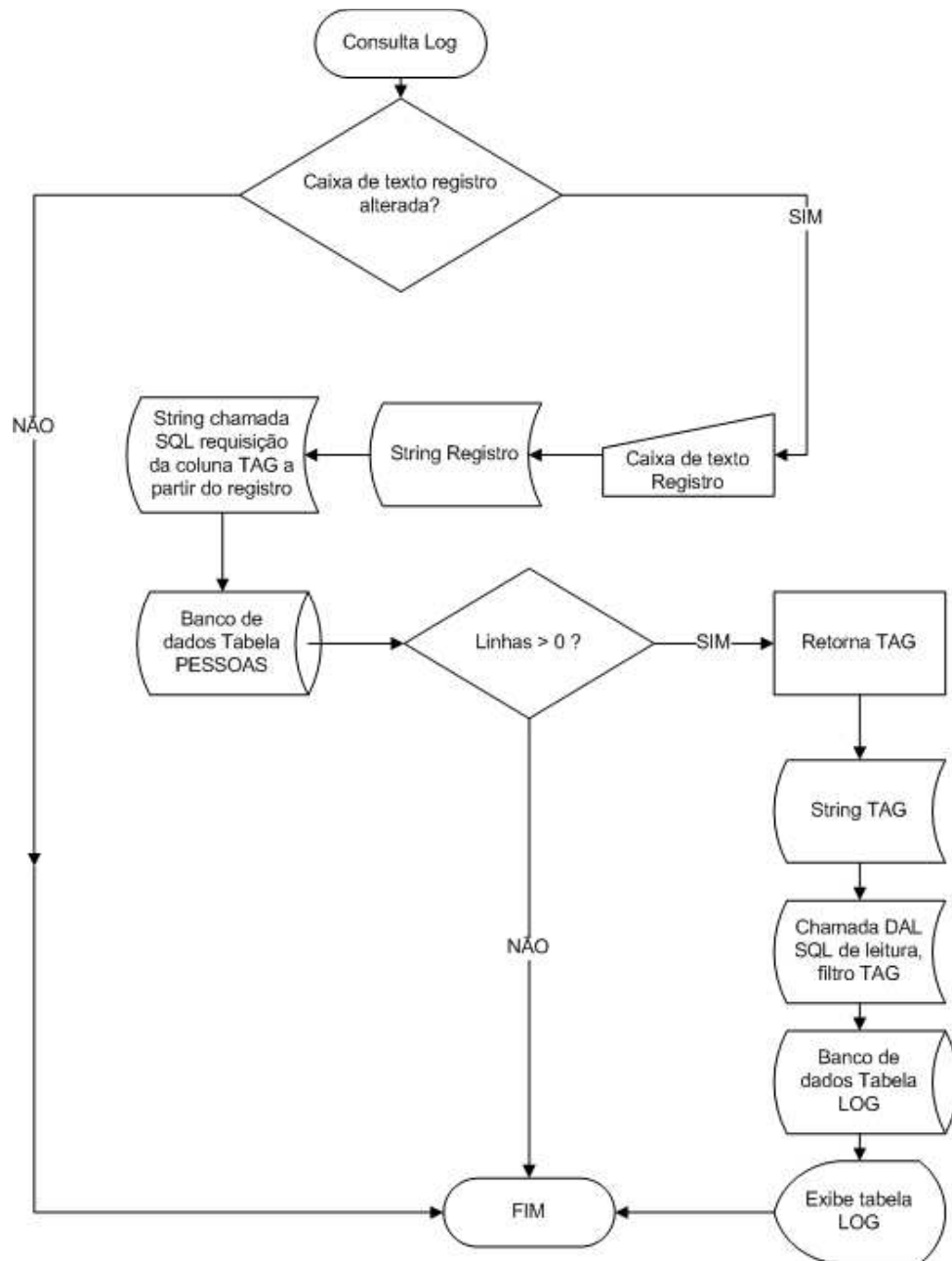
Por fim, existe a funcionalidade de controle e monitoramento de capacidade encontrada na aba consulta. É possível definir a capacidade máxima de cada bloco individualmente, e visualizar a lotação que é atualizada a cada cinco segundos. Quando se atinge a capacidade máxima o hardware indica piscando um LED amarelo, enquanto o sistema de gerencia exibe uma notificação no sistema operacional Windows. Para saber quantas pessoas estão em cada bloco as duas aplicações filtram a tabela 'PESSOAS' na coluna 'ID-BLOCO'. A capacidade máxima é armazenada e filtrada da tabela 'Blocos'.

### 3.5 TESTES DE INTEGRAÇÃO ENTRE O SISTEMA *IN LOCO* E APLICAÇÃO EM C#

A comunicação entre a interface gráfica e o sistema embarcado para o cadastro é feita no Banco de dados utilizando a tabela CATCH-TAG, Figura 15. Ao clicar no botão "Pega TAG", Figura 17, a aplicação envia para a tabela CATCH-TAG a palavra "TENTE NOVAMENTE", e abre uma janela com a mensagem PASSE A TAG que interrompe a aplicação até o usuário



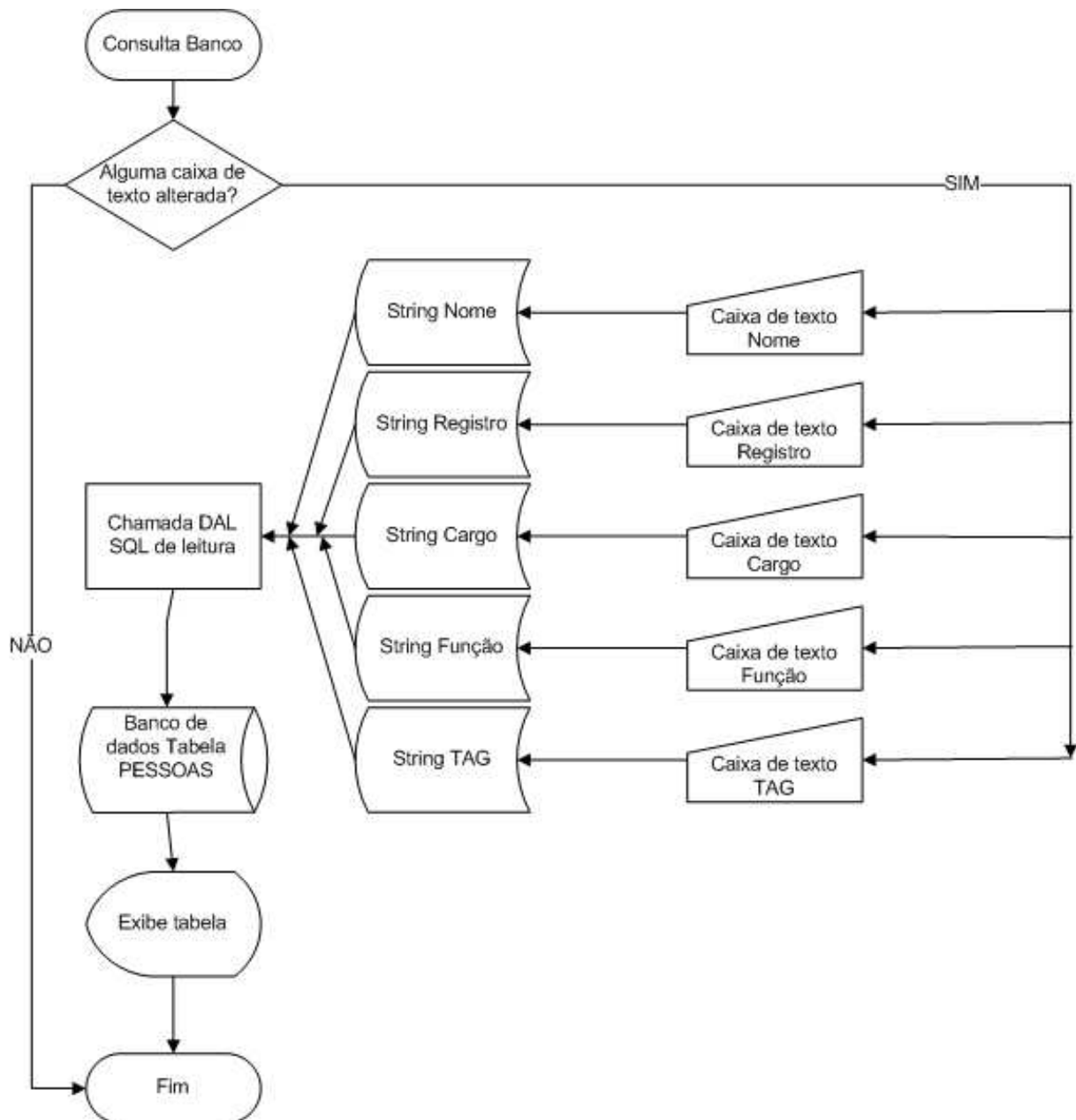
Figura 23 – Função utilizada para pesquisar log por indivíduo.



Fonte: Autoria Própria (2021).

apertar "OK". Enquanto isso o sistema embarcado estará rodando um programa que aguarda informações da TAG RFID do usuário. Ao receber as informações da TAG o sistema embarcado envia para a tabela CATCH-TAG estas informações. Após isso o usuário deverá clicar no botão "OK" e após, a aplicação realizará a leitura da última informação escrita na tabela e apresentará a mesma no campo TAG. Caso o sistema embarcado falhe em receber as informações da TAG, o sistema realizará a leitura da última linha da tabela, que será "TENTE NOVAMENTE", que foi escrita pela própria aplicação no início do processo.

Figura 24 – Função de consulta ao banco de dados.



Fonte: Autoria Própria (2021).

## 4 RESULTADOS E DISCUSSÕES

Para apresentar os resultados iniciou-se o cadastro de um indivíduo qualquer utilizando o exemplo implementado da UTFPR do campus de Campo Mourão e executou-se o código de cadastro de TAGs do sistema embarcado.

### 4.1 CADASTRO

Primeiramente executou-se a aplicação em C# em um computador de mesa com *Windows 10* que apresenta a tela da Figura 25.

**Figura 25 – Tela inicial da aplicação desenvolvida.**



**Fonte: Autoria Própria (2021).**

Ao selecionar a aba cadastro no canto superior esquerdo alternou-se para a tela da Figura 26 e adicionou-se informações fictícias de um aluno com nome de 'Aluno 01' registro 'a000001' com a função 'Eletrônica', armazenando estes dados no Banco de Dados da AWS, demonstrando que a comunicação com o banco de dados foi feita com sucesso.

Ao clicar no Botão 'Pega TAG' a caixa de mensagens da Figura 27 sobrepõe a aplicação e aguarda a interação do usuário com o botão OK. Caso a TAG não seja aproximada do sensor e a ação de clique seja efetuada o resultado obtido é como o apresentado na Figura 28 demonstrando que o software está preparado para este tipo de falha, em que o usuário não aproxima de maneira

**Figura 26 – Apresentação da funcionalidade de cadastro de usuários.**

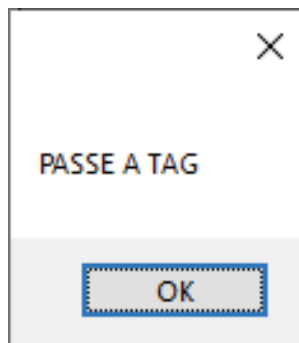
The screenshot shows a software interface with a menu bar (VISUALIZAÇÃO, CADASTRO, CONSULTA) and a date display (sábado, 24 de julho de 2021). On the left, there is a registration form with fields for Name (Aluno 01), Registro (a000001), Cargo (Aluno), Função (Eletronica), and TAG. A 'Pega TAG' button is next to the TAG field. Below the form are buttons for 'Inserir Cadastro', 'Deletar Registro', and 'Limpa Formulário'. On the right, a table displays user data:

ID	NOME	REGISTRO	CARGO	FUNÇÃO	ID_BLOCO	TAG
31	Aluno 01	a000001	Aluno	Eletronica	1	E5 4
*						

Fonte: Autoria Própria(2021).

adequada a TAG.

**Figura 27 – Caixa de mensagem para indicar ao usuário o momento de aproximar a TAG.**



Fonte: Autoria Própria (2021).

**Figura 28 – Demonstração de falha ao tentar obter TAG no cadastro.**



Fonte: Autoria Própria (2021).

Já se o administrador aproxima a TAG do sensor e pressiona OK o resultado obtido é o valor da identificação da TAG como apresentado na Figura 29, o que indica que o sistema

embarcado esta comunicando com o banco de dados também, realizando assim, a comunicação entre os dois sistemas.

**Figura 29 – Retorno na aplicação ao aproximar a TAG e pressionar OK.**



**Fonte: Autoria Própria (2021).**

Para fins de teste do protótipo, o sistema embarcado apresenta textos em tela como na Figura 30.

**Figura 30 – Resultado na interface de desenvolvimento do sistema embarcado.**

```
root@raspberrypi:/home/pi/Documents/RPi-RFID-master# ./Bora3
Conectado com sucesso no DB
INSERT INTO CATCH_TAG(id, tag ) values('0', 'E0 F4 2C 4F');
Conectado com sucesso no DB
```

**Fonte: Autoria Própria (2021).**

Por fim o administrador deve pressionar o botão 'Inserir Cadastro' para incluir os dados no banco. É possível verificar se o cadastro foi realizado clicando no botão 'Limpa Formulário' e inserindo qualquer uma das informações do indivíduo nos espaços que agora estão em branco como na Figura 31.

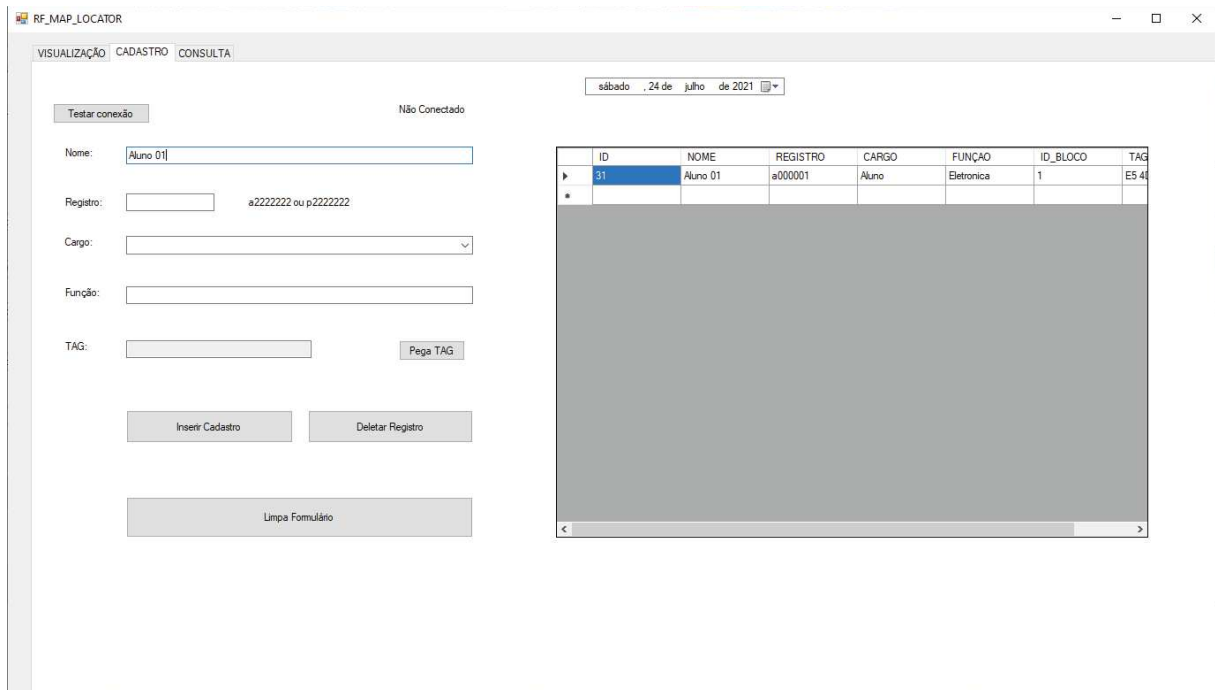
## 4.2 CONTROLE DE ACESSO E VISUALIZAÇÃO

Neste tópico foi simulada a entrada e saída do aluno cadastrado acima, no bloco A, e visualizou-se a ocupação do bloco e os registros de entrada e saída do aluno.

Primeiramente executa-se no sistema embarcado o programa para controlar a entrada de pessoas no bloco A. Aproximando uma TAG não cadastrada, para fins de teste, o sistema embarcado retorna uma mensagem informando que o aluno não está cadastrado, como na Figura 32. Também é aceso um LED vermelho indicando que o indivíduo não tem autorização para acessar o local.

Ao aproximar a TAG do aluno 01 o sistema retorna uma mensagem dizendo que o acesso foi liberado acendendo em conjunto um LED verde, simulando seu acesso ao bloco e alimentando a tabela LOG no banco de dados, demonstrando que o sistema cumpre seu papel de permitir ou negar a entrada de indivíduos.

**Figura 31 – Pesquisando usando a aba de cadastros**



Fonte: Autoria Própria (2021).

**Figura 32 – Demonstração do sistema embarcado negando a entrada de TAG não cadastrada.**

```

Conectado com sucesso no DB
currentDate () =02-05-2021
currentTime () =20:15:30
Aluno nao cadastrado
  
```

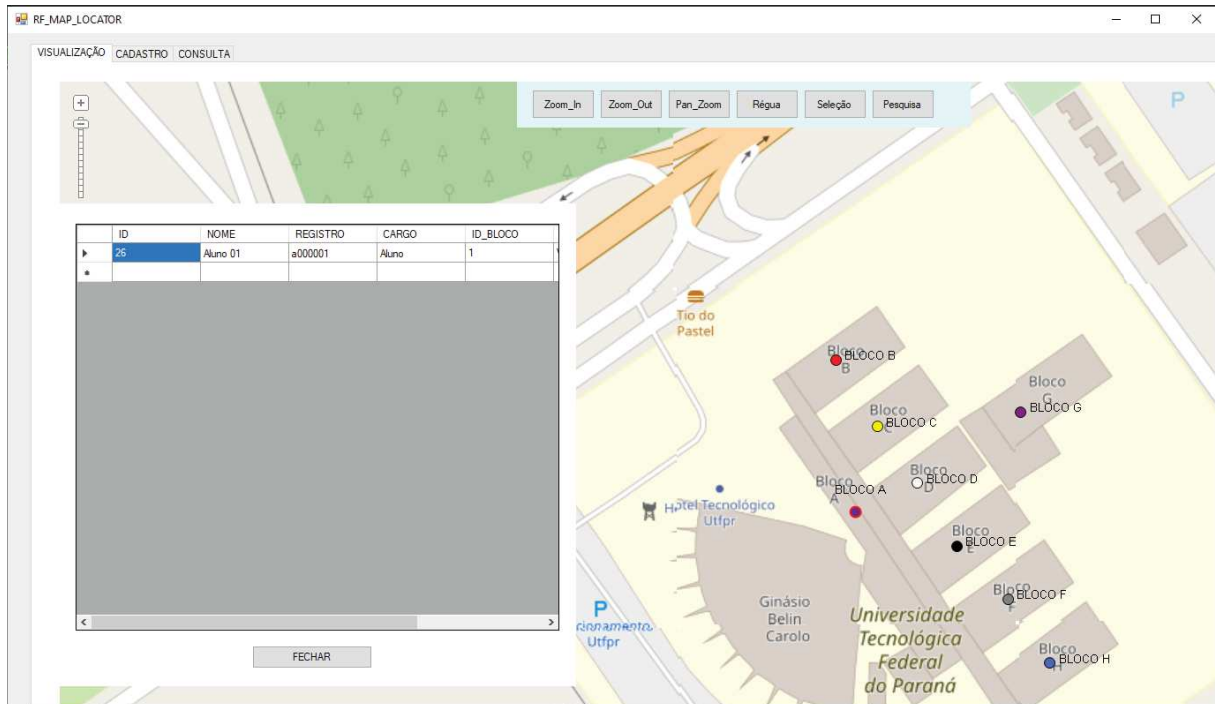
Fonte: Autoria Própria (2021).

Voltando para a visualização do aplicativo, se for realizada uma filtragem no Bloco A por alunos pode-se observar que o aluno se encontra dentro do bloco A como na Figura 33.

Ao executar no sistema embarcado o programa para controlar a saída de pessoas do bloco A e aproximar a TAG, pode-se verificar na visualização de que o aluno não consta mais dentro do bloco, neste caso específico o bloco consta como vazio. Ao selecionar no bloco a opção LOG e adicionar a data no formato da Figura 34 e pressionar Ok consta o registro de entrada e saída de alunos deste bloco na data especificada.

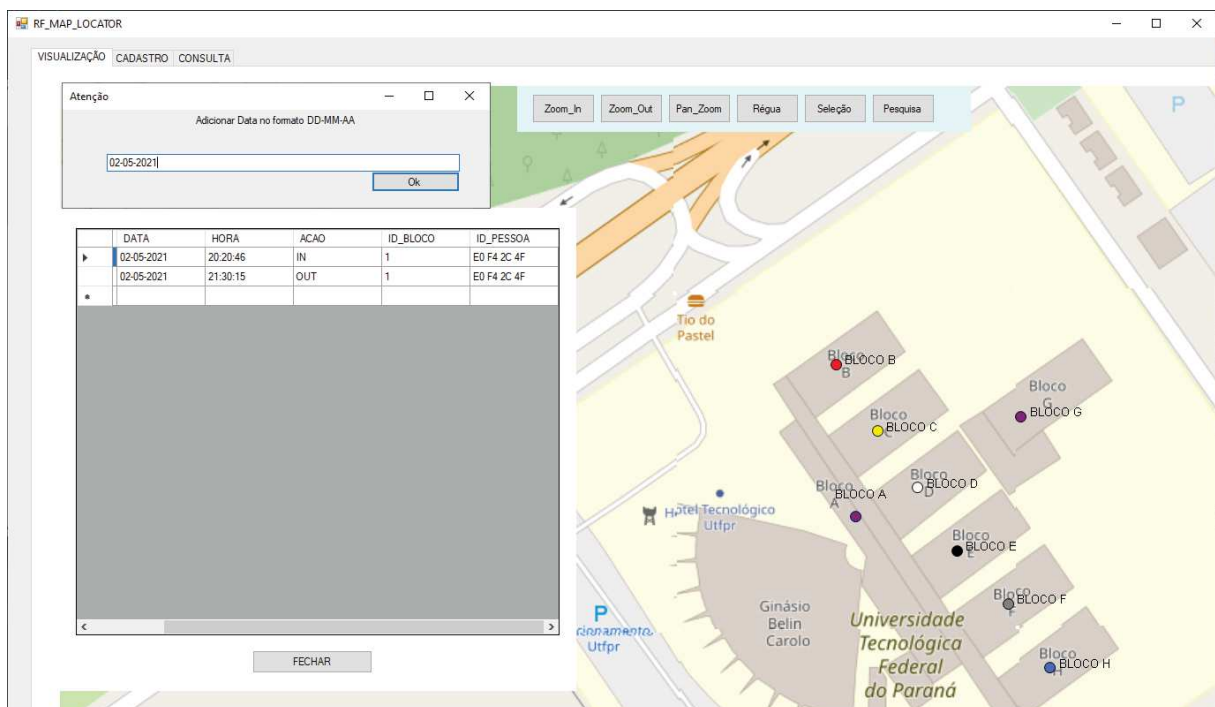
Na aba CONSULTA é possível saber todos os registros de entrada e saída dos indivíduos escolhidos, assim como realizar o controle de capacidade, como visto na Figura 35.

**Figura 33 – Demonstração do Aluno 01 dentro do bloco A**



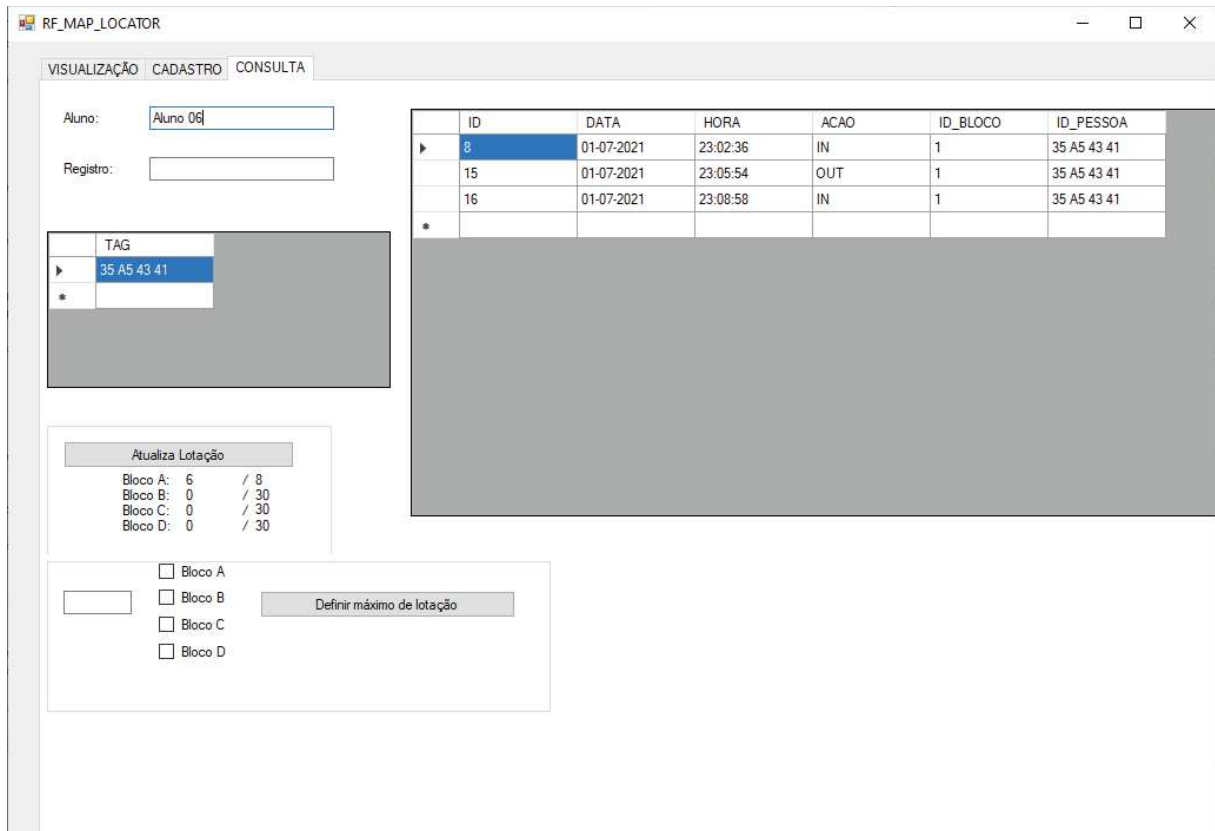
Fonte: Autoria Própria (2021).

**Figura 34 – Demonstração de como consultar o LOG do bloco A.**



Fonte: Autoria Própria (2021).

**Figura 35 – Demonstração da consulta de LOG de cada indivíduo, controle e monitoramento de capacidade.**



**Fonte: Autoria Própria (2021).**

### 4.3 CONTROLE E MONITORAMENTO DE CAPACIDADE

No canto inferior esquerdo da Figura 35 é possível observar o controle e monitoramento de capacidade. É possível observar que os blocos possuem diferentes capacidades definidas pelo administrador.

Após todos os testes realizados nesta seção é possível considerar que o software embarcado em conjunto com o *GUI* cumpre corretamente suas funções de monitorar, controlar e registrar a entrada e saída de indivíduos de determinados estabelecimentos em tempo real, sendo a principal característica a possibilidade de realizar a consulta de maneira visual através do SIG, para saber quais indivíduos se encontram em cada construção, ou quais indivíduos entraram e saíram das mesmas. Também é possível realizar a consulta filtrando cada indivíduo, fazendo-se possível realizar um rastreamento para saber por onde este indivíduo andou.

Porém, por ser um protótipo com recursos financeiros e espaços limitados não foi possível implementar um obstáculo que serviria para impedir que indivíduos não autorizados acessassem os locais propostos no teste.



Um problema que poderia ocorrer seria a perda de conexão com a internet que acarretaria na impossibilidade de se conectar com o SGBD da AWS. Se faz necessário a implementação de um outro banco de dados local para trabalhar em paralelo enquanto ocorresse a perda de conexão, possibilitando tanto consulta e escrita de ambos os módulos enquanto o sistema estiver *offline*.

No canal do autor em Projeto... (2021) é disponibilizada uma demonstração detalhada passo a passo de utilização do hardware em conjunto com o *GUI* exemplificando a entrada e saída de vários alunos em um determinado bloco.

## 5 CONCLUSÕES

Considerando o sistema desenvolvido através da integração do hardware, software embarcado e software *GUI*, foi possível verificar que as necessidades dos SACS foram atendidas através da utilização dos módulos descritos, que foram eficazes em identificar os indivíduos, controlar o acesso e quantidade de indivíduos, demonstrar em interface responsiva e realizar buscas de LOGs de entrada e saída com auxílio de SIG, sendo todos esses realizados em tempo real. Desta forma, foi concluído que esse sistema é capaz de auxiliar e atuar no controle de acesso, e assim se mostrou capaz de gerenciar diversas estruturas e sistemas como mostrados, de forma a facilitar e automatizar esses processos.

Durante o desenvolvimento outras aplicações de SACS foram identificadas com grande relevância devido ao atual cenário de pandemia, onde esse sistema pode ser adaptado e estruturado para atender as necessidades de controle de lotação e fluxo de acessos.

Este projeto se mostrou eficaz durante os estudos teóricos/práticos onde a gestão dessas informações se tornou eficiente e suficiente para o mínimo controle de um sistema SAC. Podendo atuar no bloqueio de acessos não autorizados/estranhos, rastreamento de circulação dos indivíduos, assim como outros pontos intrínsecos aos SACS.

### 5.1 TRABALHOS FUTUROS

Futuramente é possível transformar os módulos de entrada e saída do mesmo bloco, que atualmente requerem dois sistemas embarcados, em um só utilizando a premissa de que se o indivíduo já está dentro do bloco a única ação possível seria a saída. Também seria necessária uma tela de Login para apenas os usuários autorizados poderem ter acesso ao *GUI*. Outra proposta interessante, mas que requer um investimento mais elevado seria aumentar a precisão dos LOGs, fazendo com que além de registrar a entrada dos indivíduos em um bloco de uma Universidade como no exemplo usado, possa ser realizado o registro da entrada destes mesmos indivíduos nas salas de aulas, ou indo além deste exemplo, a entrada de pessoas nas lojas de um *shopping* ou até mesmo em setores de uma mesma indústria.

## REFERÊNCIAS

AL-LAWATI, A.; AL-JAHDHAMI, S.; AL-BELUSHI, A.; AL-ADAWI, D.; AWADALLA, M.; AL-ABRI, D. Rfid-based system for school children transportation safety enhancement. *In: IEEE 8TH GCC CONFERENCE AND EXHIBITION. Anais [...]. [S.l.]*, 2015. p. 1–6.

AMAZON relational database service. 2021. Disponível em: <https://aws.amazon.com/pt/rds/>. Acesso em: 23 jul. 2021.

ANATEL. **Certificado de Conformidade**. 2018. São Paulo: ANATEL, 2018. Disponível em: [https://www.raspberrypi.org/documentation/hardware/raspberrypi/compliance/rpi\\_DOC\\_3bplus\\_BRAZIL.pdf](https://www.raspberrypi.org/documentation/hardware/raspberrypi/compliance/rpi_DOC_3bplus_BRAZIL.pdf). Acesso em: 23 jul. 2021.

BENCHOFF, B. **Introducing to Raspberry PI 3**. 2016. Disponível em: <https://hackaday.com/2016/02/28/introducing-the-raspberry-pi-3/>. Acesso em: 23 jul. 2021.

CHEN, J. X. Geographic information systems. **Computing in Science Engineering**, v. 12, n. 1, p. 8–9, 2010.

CONTROLE de acesso: saiba o porquê esse processo é tão importante para a segurança. Grupo Iron, 2020. Disponível em: <https://www.grupoiron.com/blog/controle-de-acesso/>. Acesso em: 23 jul. 2021.

GALHARDO, A. T. **Sistemas eletrônicos de controle de acesso**. 2011. Monografia (Bacharelado em Engenharia Elétrica) - Universidade São Francisco, Campinas, Brasil, 2011. Disponível em: <http://lyceumonline.usf.edu.br/salavirtual/documentos/2141.pdf>. Acesso em: 23 jul. 2021.

GARCIA, H. V. Fiberworks quantum. 2021. Disponível em: <https://www.fiberworks.com.br/>. Acesso em: 23 jul. 2021.

GOMES, G. **Lotação e fluxo de pessoas em tempos de pandemia**. RFID Brasil, 2021. Disponível em: <https://rfidbrasil.com/blog/lotacao-e-fluxo-de-pessoas/>. Acesso em: 23 jul. 2021.

HEATH, S. What is an embedded system? *In: HEATH, S. (Ed.). Embedded Systems Design (Second Edition)*. Second edition. Oxford: Newnes, 2002. p. 1–14. ISBN 978-0-7506-5546-0. Disponível em: <https://www.sciencedirect.com/science/article/pii/B9780750655460500025>. Acesso em: 23 jul. 2021.

HRUSKA, J. **Windows 10 now runs on the Raspberry Pi**. 2019. Disponível em: <https://www.extremetech.com/computing/285792-windows-10-now-runs-on-the-raspberry-pi>. Acesso em: 23 jul. 2021.

KIT Módulo Leitor Rfid Mfrc522 Mifare. [201-?]. Disponível em: <https://www.filipeflop.com/produto/kit-modulo-leitor-rfid-mfrc522-mifare>. Acesso em: 6 ago. 2021.

LECHETA, R. R. **AWS para desenvolvedores: aprenda a instalar aplicações na nuvem da amazon aws**. [S.l.]: Novatec Editora, 2014. ISBN 9788575223932.

LEGHI-GARCIA, L. E. **Form1.cs**. 2021. (Código-fonte). Disponível em: [https://github.com/luksdras/UTFPR1/blob/APP\\_VISUAL/testeWFASQL/Form1.cs](https://github.com/luksdras/UTFPR1/blob/APP_VISUAL/testeWFASQL/Form1.cs). Acesso em: 6 ago. 2021.

LOVBOM, P.; BALBOA, M. **Raspberry PI library for MFRC522**. 2018. (código-fonte). Disponível em: <https://github.com/paguz/RPi-RFID>. Acesso em: 23 jul. 2021.

MADEIRA, R. **O mapa de John Snow**. 2021. Disponível em: <https://storymaps.arcgis.com/stories/886d6272a35b4560994578d792d930fe>. Acesso em: 23 jul. 2021.

MAP taken from a report by Dr. John Snow. WellcomeCollection, 2021. Disponível em: <https://wellcomecollection.org/works/uxgfjt62>. Acesso em: 23 jul. 2021.

MARINGÁ. **Decreto nº 1311/2021 de 08 de julho de 2021**. 2021. Disponível em: <http://www2.maringa.pr.gov.br/cdn-imprensa/decreto1311.pdf>. Acesso em: 23 jul. 2021.

MCCAULEY, M. **C library for Broadcom BCM 2835 as used in Raspberry Pi**. 2021. Disponível em: [www.airspayce.com/mikem/bcm2835/bcm2835-1.50.tar.gz](http://www.airspayce.com/mikem/bcm2835/bcm2835-1.50.tar.gz). Acesso em: 23 jul. 2021.

MELL, P.; GRANCE, T. **The NIST and definition of cloud and computing**. [S.l.], 2011. Disponível em: <https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-145.pdf>. Acesso em: 23 jul. 2021.

MIN, H.; XIAOYAN, X. Software development for electromagnetic tomography image reconstruction using activex. *In: 2007 8TH INTERNATIONAL CONFERENCE ON ELECTRONIC MEASUREMENT AND INSTRUMENTS. Anais [...]*. [S.l.], 2007. p. 2-559-2-562.

MYSQL Database Service. Oracle, 2021. Disponível em: <https://www.oracle.com/br/mysql/>. Acesso em: 23 jul. 2021.

NORONHA, A.; MORIARTY, R.; O'CONNELL, K.; VILLA, N. **Attaining IoT value: how to move from connecting things to capturing insights.** [S.l.], 2014. Disponível em: [https://www.cisco.com/c/dam/en\\_us/solutions/trends/iot/docs/iot-data-analytics-white-paper.PDF](https://www.cisco.com/c/dam/en_us/solutions/trends/iot/docs/iot-data-analytics-white-paper.PDF). Acesso em: 23 jul. 2021.

PROJETO de sistema de controle de entrada e saída RFID (GUI + embedded system). 2021. Publicado pelo canal Lucas Esdras. Disponível em: [https://www.youtube.com/watch?v=bc2ggkM\\_VKs](https://www.youtube.com/watch?v=bc2ggkM_VKs). Acesso em: 6 ago. 2021.

RANDALL, G. **Passive RFID tags Vs. active RFID tags.** 2017. Disponível em: [https://www.streetdirectory.com/travel\\_guide/print\\_article.php?articleId=139433](https://www.streetdirectory.com/travel_guide/print_article.php?articleId=139433). Acesso em: 23 jul. 2021.

RASPBERRY Pi 3 Model B GPIO block pinout. 2016. Disponível em: <https://www.element14.com/community/docs/DOC-73950/l/raspberry-pi-3-model-b-gpio-40-pin-block-pinout?ICID=rpimain-legacy-doc>. Acesso em: 6 ago. 2021.

RASPBERRY Pi foundation. 2014. Disponível em: <https://www.raspberrypi.org/documentation/faqs/#introduction>. Acesso em: 23 jul. 2021.

SANTA CATARINA. **Portaria SES nº 391 de 05 de junho de 2020.** 2020. Disponível em: <http://dados.sc.gov.br/dataset/0a43e611-003a-48c8-a9d4-882abcde8caa/resource/3762ce10-ec24-436d-afa7-f22c233197d3/download/portaria-ses-n-391-de-05.06.2020.pdf>. Acesso em: 23 jul. 2021.

SHAPEFILES. ESRI, 2020. Disponível em: <https://doc.arcgis.com/pt-br/arcgis-online/reference/shapefiles.htm>. Acesso em: 23 jul. 2021.

SILBERSCHATZ, A.; SUNDARSHAN, S.; KORTH, H.F. **Sistema de Banco de Dados.** [S.l.]: Elsevier, 2016. ISBN 9788535251425.

SNOW, J. **Report on the cholera outbreak in the Parish of St. James, Westminster, during the autumn of 1854.** [S.l.], 1855.

STADZISZ, R.; RENAUX, D. P. B. **Software Embarcado.** 2008. Disponível em: [https://pessoal.dainf.ct.utfpr.edu.br/douglasrenaux/Stad\\_Renaux\\_Software\\_Embarcado.pdf](https://pessoal.dainf.ct.utfpr.edu.br/douglasrenaux/Stad_Renaux_Software_Embarcado.pdf). Acesso em: 23 jul. 2021.

TEIXEIRA, T. **Controle de Fluxo de Pessoas usando RFId.** 2011. Monografia (Tecnologia em Sistemas de Telecomunicações) - Instituto Federal de Santa Catarina, São José, Brasil, 2011.

VARELA, V. **Restreamento endêmico da dengue, zika, e chikungunya via android e SIG.** 2016. Dissertação (Bacharelado em Engenharia de Redes de Comunicação) - Universidade de Brasília, Brasília, Brasil, 2016.

VARGAS, C. H. **Internet de las cosas.** 2014. Disponível em: <https://estefan345.wordpress.com/2014/03/>. Acesso em: 23 jul. 2021.

VIOLINO, B. **What is RFID?** 2005. Disponível em: <https://www.rfidjournal.com/what-is-rfid>. Acesso em: 23 jul. 2021.

WHAT is RFID? EPC-RFID INFO, 2021. Disponível em: <https://www.epc-rfid.info/rfid>. Acesso em: 23 jul. 2021.

WOO-GARCIA, R. M.; LOMELI-DORANTES, U. H.; LÓPEZ-HUERTA, F.; HERRERA-MAY, A. L.; MARTÍNEZ-CASTILLO, J. Design and implementation of a system access control by rfid. *In: 2016 IEEE INTERNATIONAL ENGINEERING SUMMIT, II CUMBRE INTERNACIONAL DE LAS INGENIERIAS (IE-SUMMIT). Anais [...]. [S.l.]*, 2016. p. 1–4.

YAMASAKI, Y.; ARITSUGI, M. A case study of iaas and saas in a public cloud. *In: 2015 IEEE INTERNATIONAL CONFERENCE ON CLOUD ENGINEERING. Anais [...]. [S.l.]*, 2015. p. 434–439.