

DESENVOLVIMENTO DE APLICATIVO PARA AUXILIAR NA LOCOMOÇÃO DE PESSOAS COM DEFICIÊNCIA VISUAL DENTRO DO IFTM-CAMPUS PARACATU

Beatriz da Silva Nascimento¹; Silas
Martins Sousa².

¹Estudante de Engenharia Elétrica, IFTM, *Campus* Paracatu, beatriz.nascimento@estudante.iftm.edu.br

²Professor do IFTM, *Campus* Paracatu, MG, silas@iftm.edu.br

Resumo: Os deficientes visuais possuem recursos que os auxiliam na movimentação em espaços externos, ou seja, em um espaço fora de uma edificação. Porém, dentro de edificações eles encontram dificuldades para se locomoverem, por exemplo, dentro de universidades. Para resolver esse problema, este trabalho desenvolve um aplicativo para auxiliar esses deficientes visuais nas dependências do Instituto Federal do Triângulo Mineiro (IFTM) em Paracatu/MG. Para isso, utilizou-se a plataforma *MIT App Inventor* para criar o aplicativo, conectado via *Bluetooth* com o microcontrolador ESP32. Além disso, foram utilizados o GPS (sistema de posicionamento global) e a tecnologia RFID (identificação por rádio frequência). Os resultados mostraram que o GPS não obteve a precisão necessária e o sistema se mostrou preciso utilizando o RFID para indicar pontos pré cadastrados por meio de *tags* RFID que foram alocadas nas portas das salas e outros locais do IFTM.

Palavras-chave: Deficiência visual; ESP32; GPS; *MIT App Inventor*; RFID.

Development of an Application to Assist in the travel of People with Visual Impairment within the IFTM-*Campus* Paracatu

Abstract: Visually impaired people have resources that help them move around in external spaces, that is, in a space outside a building. However, within buildings they find it difficult to move around, for example, within universities. To solve this problem, this work develops an application to assist these visually impaired people on the premises of the Instituto Federal do Triângulo Mineiro (IFTM) in Paracatu/MG. To do this, the *MIT App Inventor* platform was used to create the application, connected via Bluetooth with the ESP32 microcontroller. In addition, GPS (global positioning system) and RFID technology

(radio frequency identification) were used. The results showed that the GPS did not obtain the necessary precision and the system proved to be accurate using RFID to indicate pre-registered points through RFID tags that were allocated on the doors of the rooms and other locations of the IFTM.

Keywords Visual impairment; ESP32; GPS; *MIT App Inventor*; RFID

1 INTRODUÇÃO

Quando alguém pensava em tecnologia por um longo período de tempo, era comum imaginar um robô com expressões faciais tão genuínas que deixariam qualquer um impressionado (LEE, 2018). Partindo do princípio de que a tecnologia é importante na vida cotidiana, para realização de tarefas relativamente simples, percebe-se a necessidade de estudar e incentivar o desenvolvimento de novas tecnologias que possibilitem auxiliar pessoas com deficiência a executar tarefas de forma autônoma (BORGES,2021).

Segundo o censo de 2022 do IBGE (Instituto Brasileiro de Geografia e Estatística), 18,6 milhões de pessoas com dois anos ou mais, apresentam algum tipo de deficiência no Brasil. Desse total, cerca de 3,1% apresentam dificuldade para enxergar, mesmo usando óculos ou lentes de contato.

Um bom exemplo é a necessidade de se locomover dentro de um espaço desconhecido. Quando esta situação é colocada sob a perspectiva de alguém que tenha deficiência visual, esse cenário passa a ser um grande desafio (FRAUCHES,2013).

Recursos como instalação de piso tátil, placas de sinalização em braille e mapa tátil, exigem que o deficiente visual que o consultar tenha conhecimento prévio para lhes interpretar, além da necessidade de mais informações para que alguém com deficiência visual consiga locomover, utilizando apenas esses meios (ARAÚJO,2015).

Como possível alternativa para deslocamento em ambientes internos, alguns protótipos utilizando a tecnologia RFID já foram desenvolvidos. Um deles se propunha a fornecer informações a respeito de um determinado espaço, para facilitar a orientação e a navegação, visto que havia uma necessidade de maiores informações sobre o ambiente, para que a locomoção de uma pessoa com deficiência visual ocorresse de forma mais autônoma (ARAÚJO,2015).

Outro protótipo que também utiliza a comunicação por rádio frequência e o piso tátil em locais públicos, propôs fornecer informações sobre os possíveis

obstáculos, que poderiam estar no percurso e a localização do próprio indivíduo com deficiência visual seria enviada por mensagem de texto para um amigo ou familiar, para que, caso estivesse em uma situação de desorientação, o indivíduo pudesse ser localizado (BIZELLO,2016).

Alunos do Instituto Federal Fluminense, propuseram o desenvolvimento de uma bengala eletrônica que através da alocação de *tags RFID* sob o piso tátil do *campus* sinalizaria ao usuário a localização em que ele se encontrava, e permitir chegar ao local desejado (FRAUCHES et.al,2013)

Mesmo com os resultados satisfatórios obtidos por trabalhos feitos anteriormente, é possível identificar espaços para novos estudos, como utilizar outro tipo de microcontrolador para realizar a leitura das *tags*, e explorar a plataforma *MIT APP INVENTOR* que permite o desenvolvimento de aplicativo, sem a necessidade de conhecimentos profundos sobre linguagem de programação.

A dinâmica do período letivo do IFTM *campus* Paracatu, exige que os alunos se desloquem para suas respectivas salas no mínimo duas vezes por dia, pois não existem salas fixas por turma.

Diante desse cenário, a fim de contribuir para o aumento da inclusão de pessoas com deficiência visual ao meio educacional, este trabalho propõe o desenvolvimento de um protótipo voltado para o auxiliar na locomoção de estudantes com deficiência visual dentro do *campus* do Instituto Federal do Triângulo Mineiro *campus* Paracatu-MG.

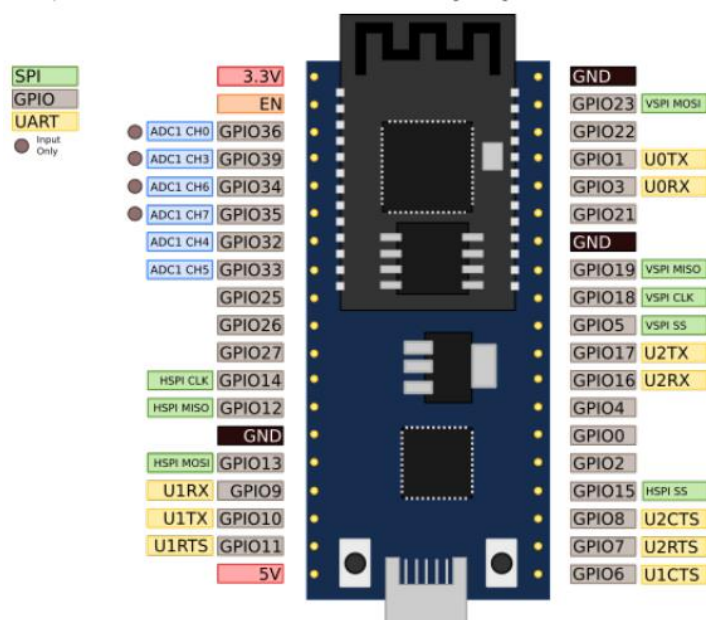
Para isso serão utilizadas *tags RFID* (etiquetas de identificação por radiofrequência), em alguns pontos da edificação e um aplicativo desenvolvido na plataforma *MIT App Inventor*, contando com ferramentas de áudio para realizar a interface com deficiente visual.

2 REFERENCIAL TEÓRICO

2.1 ESP 32

O ESP32 (figura 1), é uma plataforma microcontroladora desenvolvida pela empresa chinesa *Espressif*, tendo sua produção em massa ocorrendo no final do ano de 2016 (KOLBAN,2017).

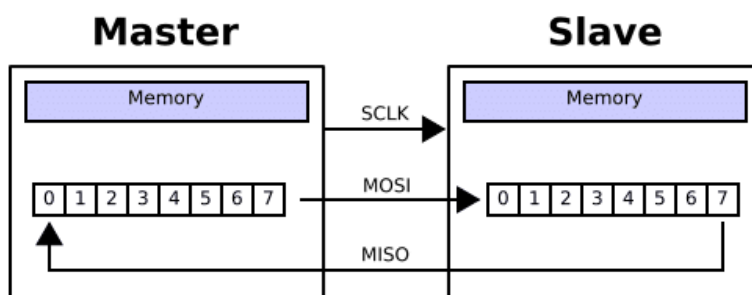
Figura 1- Esquemático ESP32.



Fonte: Kolban, 2017.

Esse dispositivo possui muitos recursos importantes, como entradas e saídas elétricas diretas (*GPIOs*) e suporte para vários protocolos, como SPI, I2C, UART. Para alimentação o ESP32 suporta 3,3V ou 5V. Ele conta também com comunicação via *Bluetooth*, que será indispensável para realização desse projeto. O protocolo de comunicação aplicado para se comunicar com o leitor RFID foi o SPI (*Serial Peripheral Interface*). Nesse protocolo utiliza-se de uma comunicação com arquitetura mestre-escravo (figura 2), em que ambos (mestre e escravo) são ligados entre si por duas linhas de dados: MOSI (*Master Output Slave*) e MISO (*Master Input Slave*). Os bits são deslocados no ritmo do sinal do relógio (SCLK) gerado pelo *master* (SACCO, 2014).

Figura 2 - Exemplo de troca de bits na comunicação SPI mestre-escravo



Fonte: SACCO, 2014.

O ESP32 foi utilizado para conectar o módulo RFID-MFRC522, fazendo a leitura do conteúdo das *tags* e posteriormente enviar as informações via *Bluetooth* para o aplicativo desenvolvido na plataforma *MIT APP INVENTOR*.

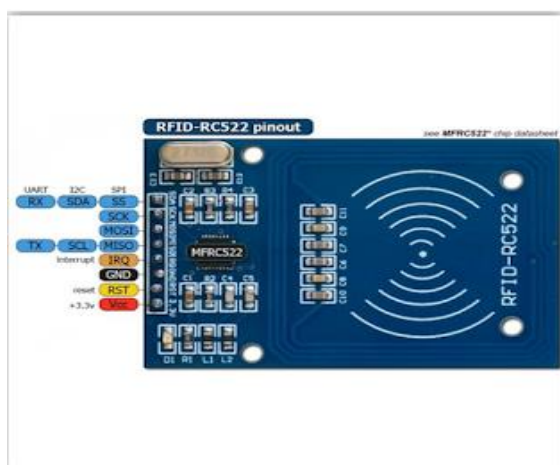
2.2 RFID

A sigla RFID é a abreviação para *Radio Frequency Identification* (Identificação por Radiofrequência). O módulo RFID-MFRC522 (figura 3) é um leitor/gravador de dados usado para comunicação sem fio em 13,56 MHz. Seu transmissor interno faz a comunicação com cartões e aparelhos emissores/receptores de mensagem, permitindo que eles respondam automaticamente a mensagens de identificação (DATASHEET, 2010, p1).

A distância para leitura e gravação entre leitor (MFRC522) e receptor (*tag*) é de no máximo 50 mm, sem antena (DATASHEET, 2010, p1).

Essa tecnologia consiste em um chip que permite a gravação de um leitor e de informações. Quando o leitor se aproxima do chip, a comunicação ocorre e as informações que foram gravadas em seguida podem ser coletadas.

Figura 3 - MÓDULO RFID-MFRC522.



Fonte: KOYANAGE, 2018.

Os chips onde os dados ficam armazenados estão dispostos em *tags* (etiquetas) ou cartões, como pode ser visto na figura 4:

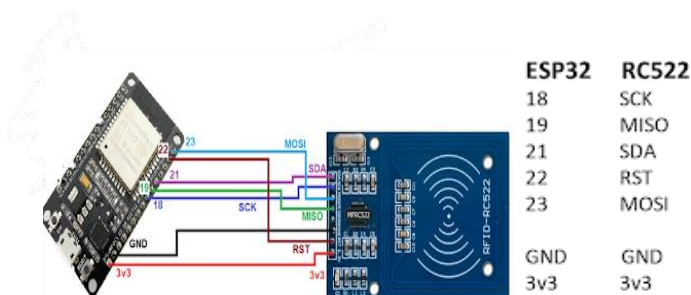
Figura 4 - *Tag*.



Fonte: Elaborado pela autora, 2023.

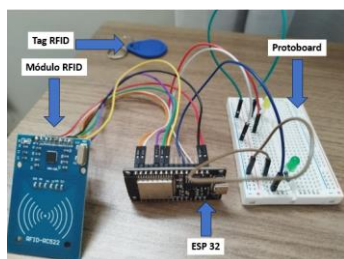
Para o desenvolvimento deste projeto foram adquiridas *tags* que possuem um número serial vindo de fábrica, esse número pode ser lido ao aproximar a *tag* do módulo RFID-MFRC522. Tendo em mãos o módulo, as *tags* e o microcontrolador, foi possível fazer a montagem da parte física do protótipo, seguindo o esquemático apresentado na figura 5 e na imagem seguinte (figura 6) pode ser visto como ficou a montagem da parte física do projeto.

Figura 5 - Esquemático de montagem da conexão entre ESP32 e MFRC522.



Fonte: KOYANAGE, 2018.

Figura 6 - Montagem da parte física conexão entre ESP32 e MFRC522.



Fonte: Elaborado pela autora, 2023.

Após a montagem da parte física para teste, o *protoboard* (placa de ensaio) foi removido juntamente com os *leds* (diodo emissor de luz), para reduzir o tamanho da caixa que vai abrigar o Esp 32 e o leitor RFID.

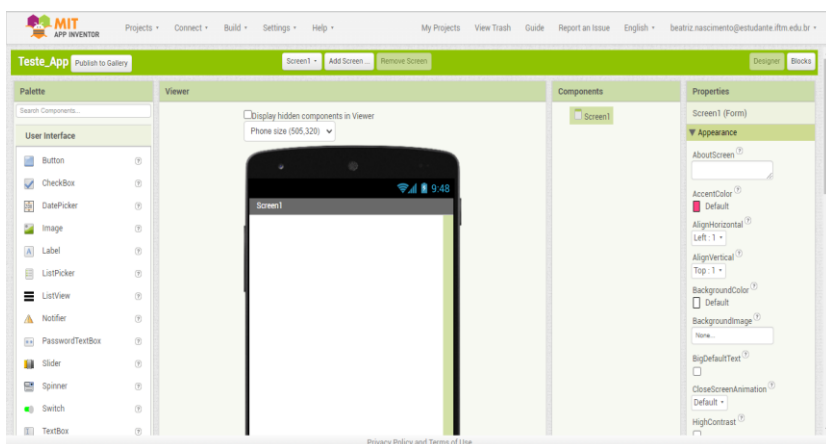
2.3 MIT APP INVENTOR

O MIT APP INVENTOR é um ambiente intuitivo para criação de aplicativos para plataformas Android ou IOS de forma gratuita. Foi desenvolvido pelo MIT (Instituto de Tecnologia de Massachusetts) juntamente com o Google (GOMES & MELO, 2013).

O App Inventor possui um ambiente de desenvolvimento baseado em blocos, entretanto, seu diferencial consiste em possibilitar aos usuários criar aplicações que incorporem serviços baseados na web, interação com redes sociais, leitura de códigos de barra, interação com sensores de orientação e geolocalização, e de funcionalidades como *text-to-speech* e reconhecimento de fala. (GOMES & MELO, 2013).

Essa é uma ferramenta que possibilita arrastar e soltar objetos criando um design (figura 7) de um aplicativo (JUNIOR & MEDINA, 2018).

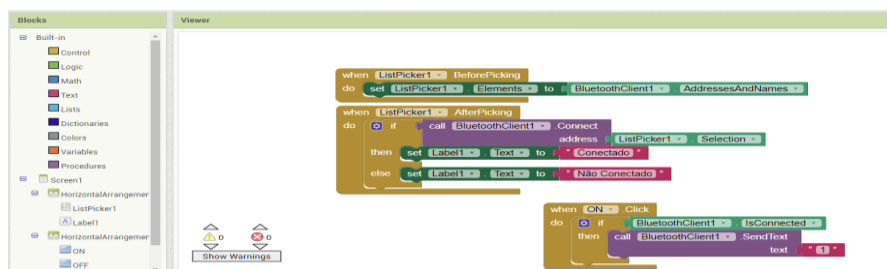
Figura 7 - MIT APP INVENTOR-Design.



Fonte: Elaborado pela autora, 2023.

Após a montagem desta montada essa etapa é possível fazer a programação do aplicativo, utilizando blocos (figura 9).

Figura 8 - MIT APP INVENTOR-Blocos.



Fonte: Elaborado pela autora, 2023.

3 MATERIAIS E MÉTODOS

Após a identificação do problema, para que houvesse a delimitação do mesmo, foi realizada uma pesquisa de materiais como artigos, teses, pesquisas que pudessem contribuir para a fundamentação teórica do assunto abordado. Posteriormente, foram adquiridos os materiais que deveriam ser utilizados para o desenvolvimento do protótipo.

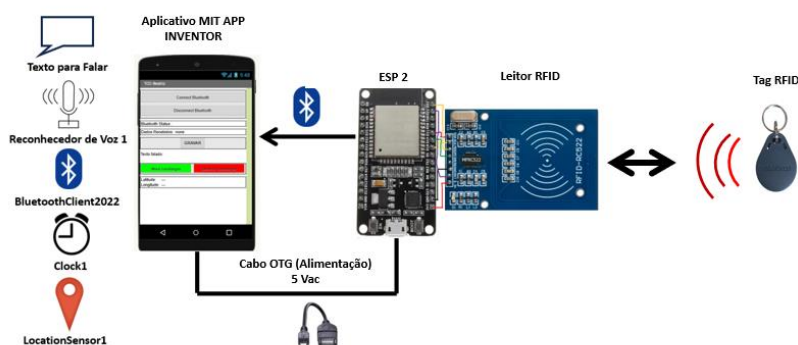
Quadro 1 - Recursos planejados para implementar o projeto.

Descrição	Unidade	Quantidade.	Valor	
			Unitário	Total
Placa de desenvolvimento ESP32	Peça	1	R\$ 38,00	R\$ 38,00
Kit Leitor RFID Cartão Tag	Unidade	1	R\$ 17,26	R\$ 17,26
Tag chaveiro com código RFID 13Mhz	Unidade	5	R\$ 6,65	R\$ 33,25
Cabo adaptador OTG	Unidade	1	R\$ 10,00	R\$ 10,00
TOTAL				R\$ 98,51

Fonte: Elaborado pela autora, 2023.

Na figura 9 pode ser visto o diagrama de funcionamento do sistema.

Figura 9 - Diagrama do Sistema



Fonte: Elaborado pela autora, 2023.

Em um primeiro momento, foi executado um código no ambiente da IDE do Arduino que leu a sequência de números presentes nas *tags*, essa sequência já vem alocada em cada uma de fábrica. No entanto, para diminuir os possíveis erros no momento de envio para o *smartphone* (celular que realiza muitas funções de um computador), optou-se por vincular cada *tag* a um código, que representaria o lugar em que estava alocada (figura 10), ou seja, esse código é o dado enviado para o *smartphone*, e permite saber a localização atual do usuário.

Figura 10 - Alocação da *tag* 5 na sala 5



Fonte: Elaborado pela autora, 2023.

A seguir estão descritos o local em que cada *tag* foi alocada, a sequência de números presente nela, e o código que deve ser enviado para o *smartphone*, assim que o usuário fizer a leitura da *tag*.

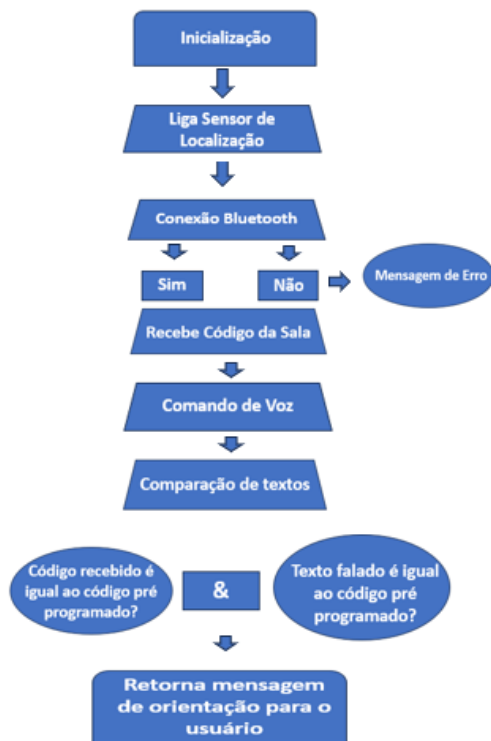
Quadro 2 – Relação da alocação e do código das *tags*.

LOCAL	SEQUÊNCIA DE NÚMEROS	CÓDIGO ENVIADO AO APLICATIVO
SALA 1	66223710	1
SALA 2	11465860	2
SALA 3	16291780	3
SALA 4	146249740	4
SALA 5	2181740	5

Fonte: Elaborado pela autora, 2023.

Após esse processo, iniciou-se a elaboração do aplicativo utilizando o *MIT APP INVENTOR*.

Figura 11 - Diagrama de Funcionamento do aplicativo.



Fonte: Elaborado pela autora, 2023.

Iniciando pela interface de interação com o usuário, onde a tela principal do aplicativo, apresentará dois botões para fazer a conexão e a desconexão via Bluetooth do *smartphone* com o microcontrolador ESP32. A informação “*Bluetooth Status*” apresenta o estado da conexão, se o dispositivo está conectado ou desconectado.

Na opção “Dados recebidos” será apresentado o código da sala que foi lido pelo módulo RFID MFRC522. O botão “Gravar”, é utilizado pelo usuário para chamar o reconhecedor de voz e informar o local para onde deseja seguir. No espaço “Texto Falado” é apresentado o texto que foi falado pelo usuário. E por fim, os dois botões “Ativar Localização” e “Desativar Localização” habilitam o sensor de localização (GPS) para determinar qual a latitude e longitude do local onde usuário se encontra.

Indo para a parte de programação de blocos, destacam-se aqui, os blocos principais para o desenvolvimento do aplicativo, sendo eles: o bloco responsável pela conexão via *Bluetooth*, seguido pelo bloco responsável por receber os dados enviados pelo microcontrolador ESP32, sendo seguido pelo bloco que captura o áudio falado pelo usuário e retorna à instrução sobre qual direção seguir.

O bloco denominado “ListPickerConnectBluetooth” (Figura 12) é o bloco que faz a conexão com o endereço Bluetooth escolhido. Esse bloco se divide em duas partes, denominadas “Antes de Escolher” e “Depois de Escolher.” Dentro da opção “Antes de Escolher”, o bloco verifica se o Bluetooth está ativo, e caso esteja abre uma lista com endereços e nomes de *bluetooth’s* disponíveis para que o usuário possa selecionar em

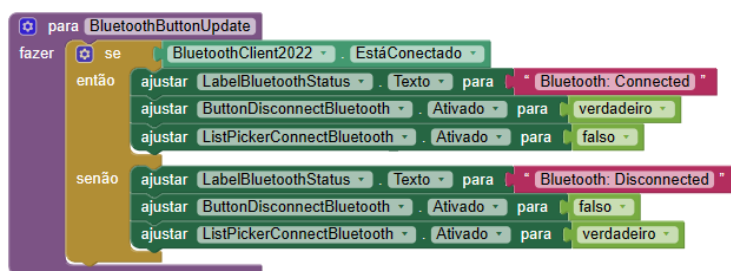
qual deseja se conectar. Caso o *Bluetooth* não esteja ativado, aparece uma mensagem na tela principal e também na tela que apareceria os nomes com endereços, avisando ao usuário que o Bluetooth não recebeu permissão para se conectar. No bloco “Depois de Escolher”, aparece a lista com bluetooth’s disponíveis e em seguida chama o bloco de atualização de status do *Bluetooth* denominado “*Bluetooth Button Updade*” (Figura 13), esse bloco altera o status de *bluetooth* para conectado ou desconetado, e imprime a mensagem na tela para o usuário, com o status atual.

Figura 12 - Bloco *ListPickerBluetooth*.



Fonte: Elaborado pela autora, 2023.

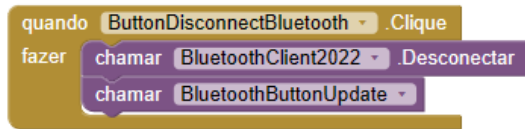
Figura 13 - Bloco *Bluetooth Button Updade*.



Fonte: Elaborado pela autora, 2023.

O Bloco denominado “*ButtonDisconnectBluetooth*” (Figura 14) é responsável por desconectar o Bluetooth e também chamar o bloco de atualização de status, que vai realizar a mesma ação descrita anteriormente, que seria imprimir uma mensagem para o usuário com o status atualizado.

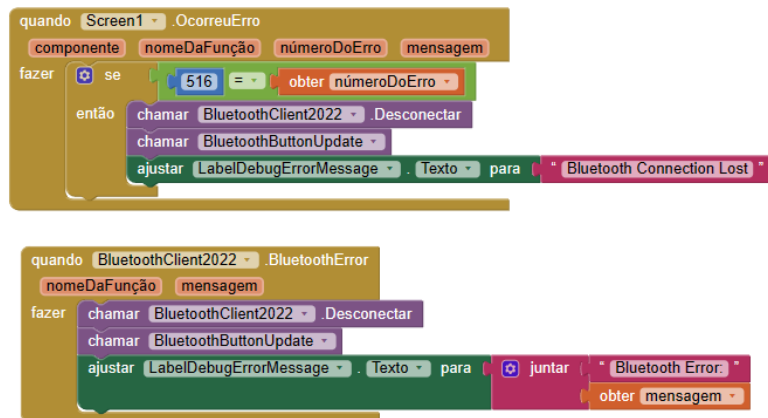
Figura 14 - Bloco *ButtonDisconnectBluetooth*.



Fonte: Elaborado pela autora, 2023.

Os blocos “*Screen1. Ocorreu Erro*” e *BluetoothCliente2022.Bluetooth Error*” (Figura 15) são responsáveis por conferir a existência de possíveis erros, ou de perda de conexão quando o aplicativo é inicializado e no momento de conexão com o endereço Bluetooth que foi selecionado.

Figura 15. Blocos *Screen1. Ocorreu Erro* e *BluetoothCliente2022.Bluetooth Error*.



Fonte: Elaborado pela autora, 2023.

O bloco “*CLOCK1*” (Figura 16) é responsável por receber as informações que são enviadas pelo ESP32, nesse caso, ele recebe o código da *tag*, que foi lida pelo módulo RFID MFRC522. Então primeiro ele verifica se o Bluetooth está conectando e se há *bytes* para receber, então ele testa se os *bytes* que vai receber são maiores que zero. Caso todas as afirmações anteriores sejam verdadeiras, o bloco inicializa uma variável local que armazena esse *byte* recebido e o imprime na tela do usuário

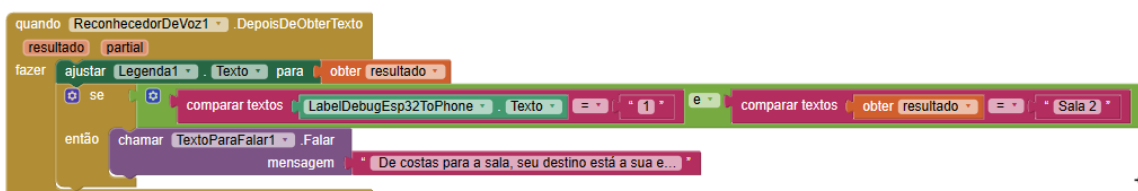
Figura 16 - Bloco CLOCK1.



Fonte: Elaborado pela autora, 2023.

Quando o usuário pressiona o botão “Gravar” o bloco “Reconhecedor de voz1” (Figura 17) é acionado, dentro dele, são comparados alguns textos. Primeiramente é feita a comparação do texto que foi enviado pelo ESP32, para verificar se o texto recebido corresponde ao código que foi estipulado para uma sala. E em seguida o texto obtido pelo reconhecedor de voz (comando do usuário) e o nome do destino, são comparados, dessa forma, se as informações forem verdadeiras, o bloco chama a função “Texto para Falar1” que retorna ao usuário a orientação sobre qual direção seguir.

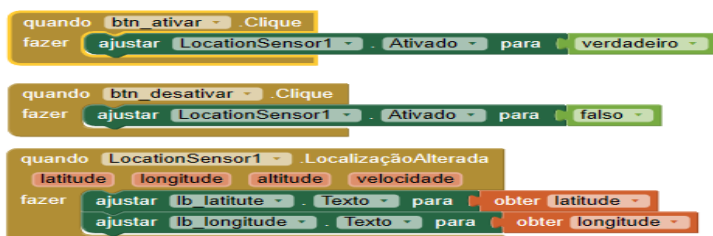
Figura 17. Bloco Reconhecedor de voz1.



Fonte: Elaborado pela autora, 2023.

Os blocos que se seguem, “btn_ativa” e “btn_desativar” habilitam ou desabilitam o sensor de localização (figura 18). E após ser habilitado, o bloco “LocationSensor1” captura a latitude e longitude da localização atual do usuário e imprime os valores na tela principal.

Figura 18 - Bloco btn_ativa e btn_desativar.



Fonte: Elaborado pela autora, 2023.

4 RESULTADOS E DISCUSSÃO

Os primeiros testes foram realizados com a ferramenta denominada sensor de localização, presente na plataforma *MIT APP INVENTOR* que utiliza do GPS (Sistema de Posicionamento Global) e retorna as coordenadas geográficas, latitude e longitude. A coleta desses valores tem por finalidade comparar os valores de latitude e longitude da localização do destino do usuário. O GPS pode apresentar um erro de até 3 metros, no entanto como a coleta de dados é feita em uma edificação, o erro pode ser maior.

Assim, com o *MIT App Inventor*, foram coletados os dados de latitude e longitude das salas utilizadas para o desenvolvimento do projeto, e esses mesmos dados coletados foram comparados com os que são fornecidos pelo aplicativo de GPS do Google, tendo como finalidade verificar se os dados coletados pelo APP INVENTOR são coerentes. Os dados coletados podem ser observados e comparados na figura 19.

Figura 19 - Medição Latitude e Longitude

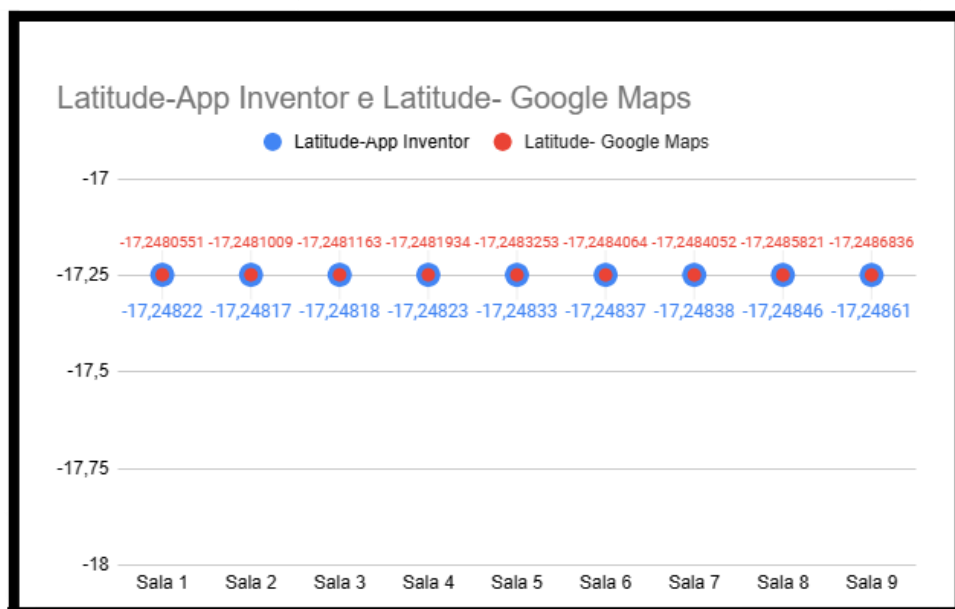
Salas	Latitude (App Inventor)	Latitude (Google Maps)	Longitude (App Inventor)	Longitude (Google Maps)
1	-17.2482	-17.2480551	-46.88096	-46.8808672
2	-17.24817	-17.2481009	-46.88095	-46.8809420
3	-17.24818	-17.2481163	-46.88094	-46.8809457
4	-17.24823	-17.2481934	-46.88097	-46.8811643
5	-17.24833	-17.2483253	-46.88108	-46.8810352
6	-17.24837	-17.2484064	-46.88107	-46.811297
7	-17.24838	-17.24840525	-46.88114	-46.8811844
8	-17.24846	-17.2485821	-46.88124	-46.8813014
9	-17.24861	-17.2486836	-46.88146	-46.8815099

Fonte: Elaborado pela autora, 2023.

O *MIT APP INVENTOR*, apresenta valores com menos casas quando comparado com os valores apresentados pelo Google *Maps*, essa configuração não pôde ser alterada.

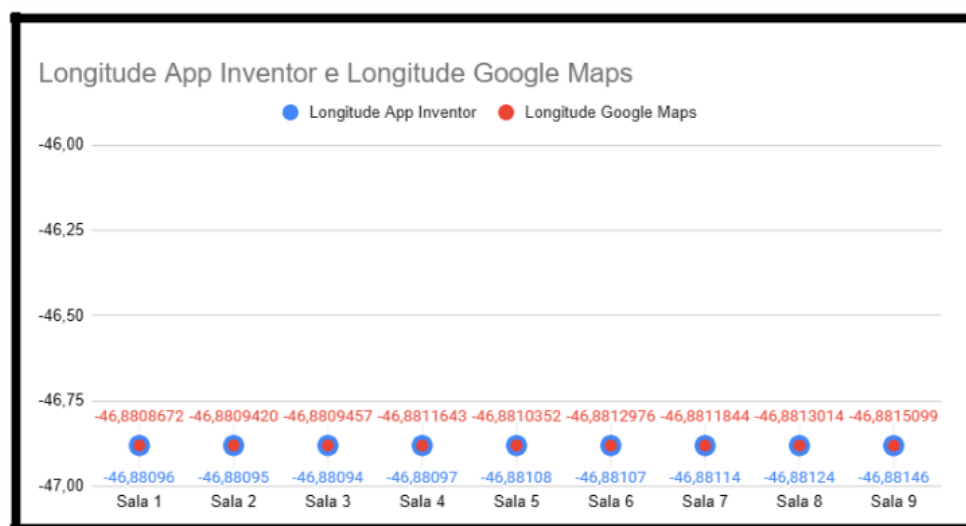
Após a coleta dos dados o objetivo seria fazer a diferença entre os valores de latitude e longitude. Essa comparação avaliaria o aumento ou diminuição desses valores, e com isso poderia informar ao usuário a direção que ele deve seguir para chegar ao destino solicitado. Os dados podem ser analisados de uma melhor forma nos gráficos a seguir.

Figura 20 - Medição Latitude



Fonte: Elaborado pela autora, 2023.

Figura 21 - Medição Longitude



Fonte: Elaborado pela autora, 2023.

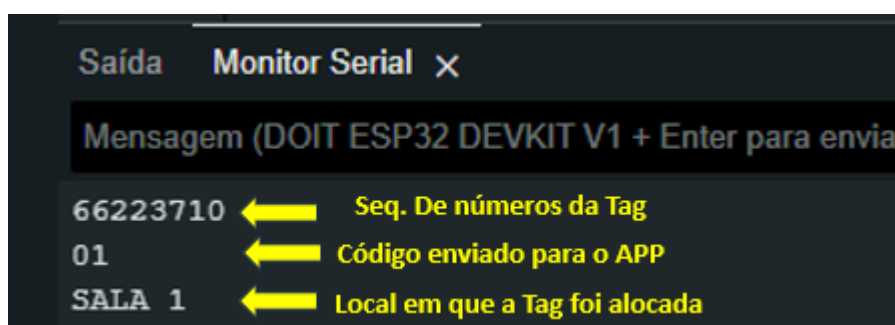
No primeiro gráfico (figura 20) tem-se os valores de latitude coletados, e o gráfico seguinte (figura 21), são os dados de longitude, coletados pelo Google Maps e também pelo MIT APP INVENTOR.

Como é possível perceber, ao analisar os valores coletados, a diferença entre eles é muito pequena, tanto de latitude quanto de longitude, o que poderia gerar erros no momento da execução do projeto. Com esses dados também foi possível constatar o que foi afirmado por Rodrigues (2011), sobre o deslocamento e orientação que ocorre em ambientes internos utilizando o GPS, pode apresentar fragilidade de sinal.

Após essas ponderações, foi decidido uma nova abordagem para a continuação do projeto, utilizando a alocação de *tags* RFID.

Para os testes feitos com as *tags*, verificou-se a leitura de cada uma delas no ESP32, que ao aproximar o leitor de uma *tag* disposta em uma das salas, obteve-se o resultado apresentado na figura 22.

Figura 22 - Informações das *tags* apresentadas na interface da IDE Arduino.



Fonte: Elaborado pela autora, 2023.

Com os testes realizados no aplicativo foi possível verificar o funcionamento da conexão Bluetooth com o ESP 32, e a recepção dos dados que foram enviados também pelo ESP32 (Figura 23).

Figura 23: Tela do Aplicativo



Fonte: Elaborado pela autora, 2023.

Realizando os testes, com a comunicação *bluetooth* entre o *Smartphone* e o ESP32 observou-se que é necessário ficar atento as permissões que cada aparelho *smartphone* requer para os aplicativos que são instalados nele. No caso do aplicativo desenvolvido por este projeto e no *smartphone* em que ele foi testado, foi necessário dar permissões para o aplicativo ter acesso ao microfone, a localização e a dispositivos que estejam próximos a ele (Figura 24). O modelo de *smartphone* utilizado foi um POCO X5, da fabricante chinesa XIOMI.

Figura 24 - Tela do *smartphone* com as Permissões do Aplicativo



Fonte: Elaborado pela autora, 2023.

Ainda sobre a conexão Bluetooth, verificou-se que é importante realizar previamente o pareamento entre o *smartphone* e o ESP32

Também foram testados os sensores utilizados no *MIT APP INVENTOR*. O comando de voz conseguiu captar o que foi solicitado, e a mensagem de áudio com a orientação para o usuário, também foi bem-sucedida, sendo reproduzida de forma audível e compreensível.

No entanto, é importante deixar claro, que para a realização desse projeto, foram utilizadas um total de cinco *tags* e todas foram alocadas em apenas um trecho do Instituto, portanto, as mensagens de voz com instrução que serão passadas para o usuário podem parecer semelhantes.

O aplicativo não foi desenvolvido no intuito de calcular uma rota instantaneamente a medida em que o usuário se locomove, mas de apresentar uma rota pré-definida para orientar o usuário que deseja se locomover de um ponto a outro.

Como citado por Frauches (2013), percebeu-se a necessidade uma melhoria técnica devido ao sensor necessitar que a tag se aproxime bastante para poder realizar a leitura.

5 CONCLUSÃO

Com o desenvolvimento desse projeto foi possível perceber como a tecnologia pode auxiliar na inclusão de pessoas com deficiência visual, proporcionando sua autonomia.

Também foi possível adquirir conhecimento sobre a plataforma *MIT APP INVENTOR*, e a programação com blocos, que é uma ótima ferramenta, mesmo para pessoas que não tenham muita experiência com programação.

O estudo também possibilitou ampliar o conhecimento em relação a placa ESP32, desde a parte de como essa placa pode ser alimentada até o envio de dados via *Bluetooth*, que foi de suma importância para a conclusão desse projeto.

Sugestão para Trabalhos Futuros

Como sugestão para projetos futuros, pode-se utilizar dos valores de latitude e longitude de cada local, e fazer uma subtração entre os valores do local atual e os valores do local para onde se deseja ir, assim com a informação em mãos, o aplicativo avalia se os novos valores são maiores ou menores que o atual, permitindo assim, saber a direção do destino solicitado pelo usuário.

Outra sugestão seria fazer o envio de dados colhidos pelo leitor RFID por meio do cabo OTG, assim, não seria necessário o pareamento do *Bluetooth*, que poderia ser um obstáculo para pessoas com deficiência visual.

REFERÊNCIAS BIBLIOGRÁFICAS

ARAÚJO, Renato P. Sistema RFID de piso tátil para localização de deficientes visuais em ambientes fechados. 2015. 62 f. Dissertação (Mestrado em Engenharia Biomédica)- Pontifícia Universidade Católica, 2015. Disponível em: [Sistema RFID de piso tátil para localização de deficientes visuais em ambientes fechados.](#) . Acesso em: 05/10/2023.

BIZELLO, Juliano S. Piso tátil com tecnologia NFC para acessibilidade de deficientes visuais em ambientes públicos. 2015. 54p. Trabalho de Conclusão de Curso. (Tecnólogo em redes de computadores) -Universidade Federal de Santa Maria, 2016. Disponível em: [Piso tátil com tecnologia NFC para acessibilidade de deficientes visuais em ambientes públicos"](#) . Acesso em: 05/10/2023.

BORGES, Wanessa; MENDES, Eniceia G. Recursos de Acessibilidade e o Uso dos Dispositivos Móveis como Tecnologia Assistiva por Pessoas com Baixa Visão. Disponível em: <https://www.scielo.br/j/rbee/a/C4GxYprjw5KMcTYB3nfcXSs/#> . Acesso em: 05/10/2023.

FRAUCHES, Ane et al. Bengala inteligente: um sistema de baixo custo para locomoção autônoma de pessoas com deficiência visual. Disponível em: [ce8497c381737baea6ad35fd6b8ae0e0.pdf \(sistemaolimp.org\)](#) . Acesso em: 05/10/2023.

GOMES, Tancicleide; MELO, Jeane. *MIT APP INVENTOR for Android*: Uma Nova Possibilidade para o Ensino de Lógica de Programação. Disponível em: [MIT MIT APP INVENTOR for Android: Uma Nova Possibilidade para o Ensino de Lógica de Programação](#) . Acesso em: 07/10/2023.

IBGE-Instituto Brasileiro de Geografia e Estatística. CENSO 2022. Disponível em: [Apresentação - Pnad Contínua - Pessoas com Deficiência 2022](#) . Acesso em: 01/10/2023.

JUNIOR, Jorge Alberto Fener Flores. Ferramenta de desenvolvimento de jogos: uma comparação entre APPSGEYSER e MIT *MIT APP INVENTOR* MIT. 2018, 25 f. Artigo de Conclusão de Curso (Especialista em tecnologias da informação e da comunicação

aplicadas a educação) – Universidade Federal de Santa Maria, 2018. Disponível em: [Jorge Alberto Fenner Flores Junior](#) . Acesso em: 07/10/2023.

KOLBAN, Neil. *Kolban's book on ESP32*. Disponível em: [kolban-ESP32.pdf](#). Acesso em: 12/10/2023.

KOYANAGE, Fernando. ESP32 com RFID: Controle de Acesso. Disponível em: [ESP32 com RFID: Controle de Acesso - Fernando K Tecnologia](#). Acesso em: 29/11/2023.

LEE, Kai-Fu. Inteligência Artificial: Como os robôs estão mudando o mundo, a forma como, amamos, nos relacionamos, trabalhamos e vivemos. Tradução: Marcelo Barbão – 1 ed. Rio de Janeiro, 2019. *E-book*. 280p. Disponível em: https://edisciplinas.usp.br/pluginfile.php/6549307/mod_resource/content/1/Inteligencia%20artificial%20-%20Kai-Fu%20Lee.pdf .Acesso em 29/11/2023.

MOTTA, Allan. O que é Arduino e como funciona. Disponível em: [O que é Arduino? | Entenda o que é e como funciona | Portal VDS \(vidadesilicio.com.br\)](#) . Acesso em: 29/11/2023.

NXP_SEMICONDUCTORES. *Datasheet* RFID MFRC522. Disponível em: < [Datasheet MFRC522](#) >. Acesso em: 08/10/2023.

RODRIGUES, Moisés. Localização em ambientes internos utilizando múltiplas tecnologias sem fio. 2011. 145 f. Dissertação (Pós- Graduação em Ciência da Computação) – Universidade Federal de Minas Gerais, 2018. Disponível em: [Repositório Institucional da UFMG: Localização em ambientes internos utilizando múltiplas tecnologias sem fio](#) . Acesso em: 02/10/2023.

SACCO, Francesco. Comunicação SPI-Parte 2. Disponível em: [Comunicação SPI – Parte 2 - Saiba como implementar - Embarcados](#) . Acesso em: 29/11/2023.