

UNIVERSIDADE DE SÃO PAULO

Instituto de Ciências Matemáticas e de Computação

Bandeja Inteligente com Rastreamento de Pessoas e de
Objetos

Antônio Pedro Lavezzo Mazzarolo



São Carlos – SP

Bandeja Inteligente com Rastreamento de Pessoas e de Objetos

Antônio Pedro Lavezzo Mazzarolo

***Orientador:* Prof. Dr. Fernando Santos Osório**

Monografia final de conclusão de curso apresentada ao Instituto de Ciências Matemáticas e de Computação – ICMC-USP, como requisito parcial para obtenção do título de Bacharel em Computação.
Área de Concentração: Sistemas Embarcados

USP – São Carlos
Junho de 2017

Mazzarolo, Antônio Pedro Lavezzo

Bandeja Inteligente com Rastreamento de Pessoas e de
Objetos / Antônio Pedro Lavezzo Mazzarolo. - São Carlos -
SP, 2017.

43 p.; 29,7 cm.

Orientador: Fernando Santos Osório.

Monografia (Graduação) - Instituto de Ciências
Matemáticas e de Computação (ICMC/USP), São Carlos -
SP, 2017.

1. QR Code. 2. RaspberryPI. 3. Localização
Indoor. 4. Administração de Medicamentos. 5. Bandeja
Inteligente. I. Osório, Fernando Santos. II.
Instituto de Ciências Matemáticas e de Computação
(ICMC/USP). III. Título.

AGRADECIMENTOS

RESUMO

MAZZAROLO, ANTÔNIO P. L.. **Bandeja Inteligente com Rastreamento de Pessoas e de Objetos**. 2017. 43 f. Monografia (Graduação) – Instituto de Ciências Matemáticas e de Computação (ICMC/USP), São Carlos – SP.

No cenário atual da medicina existe uma grande demanda de pacientes em hospitais para serem atendidos por poucos médicos e enfermeiros, existindo uma sobrecarga no trabalho dos mesmos, de modo que isto dificulta o correto atendimento e distribuição de medicamentos aos pacientes. Um erro na correta aplicação da prescrição médica pode ser muito perigoso ou até mesmo ser fatal. A proposta desse trabalho foi desenvolver uma bandeja inteligente (para medicamentos) utilizando uma RaspberryPI que oriente o profissional de saúde, identificando os remédios transportados, os pacientes que devem receber os medicamentos, e profissionais autorizados/responsáveis por realizar este procedimento, mantendo um relatório de suas atividades e auxiliando para evitar problemas como dupla medicação, medicação indevida e interação medicamentosa. Este sistema opera por meio de QR Codes, bem como o seu posicionamento considera o sinal das redes Wi-Fi para rastreamento da sua localização por meio da técnica de *fingerprinting*, indicando ao responsável onde se encontra a bandeja para evitar confusão de pacientes na medicação. Com isto, busca-se então, de forma menos custosa, solucionar ou reduzir os problemas referentes ao erro humano no atendimento de pacientes em hospitais, garantindo a correta atribuição e execução de tarefas.

Palavras-chave: QR Code, RaspberryPI, Localização Indoor, Administração de Medicamentos, Bandeja Inteligente.

ABSTRACT

MAZZAROLO, ANTÔNIO P. L.. **Bandeja Inteligente com Rastreamento de Pessoas e de Objetos**. 2017. 43 f. Monografia (Graduação) – Instituto de Ciências Matemáticas e de Computação (ICMC/USP), São Carlos – SP.

Key-words: QR Code, RaspberryPI, Indoor Localization, Medical Management, Smart Tray.

LISTA DE ILUSTRAÇÕES

Figura 1 – Exemplo da estrutura de um QR Code.	24
Figura 2 – Exemplo da organização do circuito da Bandeja	27
Figura 3 – Dados da leitura do sinal Wi-Fi	31
Figura 4 – Foto da Bandeja	41
Figura 5 – Foto do circuito da Bandeja	41
Figura 6 – Exemplo da execução correta da localização Wi-Fi	41
Figura 7 – Exemplo da execução correta da leitura do QR Code	42
Figura 8 – Exemplo da execução falha da leitura do QR Code	42

LISTA DE CÓDIGOS-FONTE

Código-fonte 1 – Script que lê o QR Code	29
Código-fonte 2 – Script que determina a localização da bandeja	29
Código-fonte 3 – Programa que gerencia o sistema	37
Código-fonte 4 – Programa que determina a localização segundo as leituras do FIND .	39

LISTA DE ABREVIATURAS E SIGLAS

AoA	Angle of Arrival
FIND	Framework for Internal Navigation and Discovery
GPS	Global Positioning System
GSM	Global System for Mobile Communications
ICMC	Instituto de Ciências Matemáticas e de Computação
IoT	Internet of Things
ISO	Organização Internacional de Padronização
LED	Light Emitting Diode
NFC	Near Field Communication
QR Code	.	Quick Response Code
RFID	Radio-Frequency Identification
RSSI	Received Signal Strength Indication
SVM	Support Vector Machine
ToA	Time of Arrival
USP	Universidade de São Paulo

SUMÁRIO

1	INTRODUÇÃO	21
1.1	Contextualização e Motivação	21
1.2	Objetivos	22
1.3	Organização da Monografia	22
2	REVISÃO BIBLIOGRÁFICA	23
2.1	Considerações Iniciais	23
2.2	Conceitos e Técnicas Relevantes	23
2.2.1	<i>QR Code</i>	23
2.2.2	<i>Localização Indoor</i>	24
2.3	Trabalhos Relacionados	25
2.4	Considerações Finais	25
3	DESENVOLVIMENTO	27
3.1	Considerações iniciais	27
3.2	Descrição Geral de Funcionamento do Sistema	27
3.3	Descrição das Atividades Realizadas	28
3.3.1	<i>Sistema Eletrônico</i>	28
3.3.2	<i>Sistema de leitura de QR Code</i>	29
3.3.3	<i>Sistema de Localização via Wi-Fi</i>	29
3.4	Resultados Obtidos	31
3.5	Dificuldades, Limitações e Trabalhos Futuros	31
3.6	Considerações Finais	32
4	CONCLUSÃO	33
4.1	Considerações Sobre o Projeto Desenvolvido	33
4.2	Considerações sobre o Curso de Graduação	33
	REFERÊNCIAS	35
APÊNDICE A	CÓDIGOS UTILIZADOS	37
APÊNDICE B	TESTES E MATERIAL UTILIZADO	41

ANEXO A DOCUMENTAÇÃO DE BIBLIOTECAS 43

INTRODUÇÃO

1.1 Contextualização e Motivação

Os sistemas embarcados, âmbito no qual esse projeto se situa, se dedicam à executar atividades específicas para obter um melhor desempenho e evitar falhas em sistemas críticos, presentes, por exemplo, em sistemas de controle de acesso biométrico, equipamentos portáteis de medição, equipamentos de rede e sistemas de monitoramento médico. Nesse último caso, encontramos dispositivos não tolerantes a falhas e dispositivos contra falhas, ou seja, sistemas que diminuem a possibilidade de erro humano e facilitam o trabalho do profissional apto a tomar decisões a partir dessa informação (MALONE; SAVERNO, 2012).

O monitoramento do tratamento de pacientes, não só através de sensores, estímulos físicos e reações a determinados procedimentos, é feito também pela correta medicação do mesmo, além do rápido atendimento em estados críticos. Para garantir o atendimento correto dos mesmos, é preciso gerenciar modos de identificação rápida da pessoa, do local e do problema que deve ser tratado, ou seja, a localização precisa dos profissionais de saúde e das pessoas em cuidado é muito importante, assim como a localização e correta aplicação dos medicamentos utilizados por estes (LIN *et al.*, 2006).

As redes Wi-Fi existem em praticamente todos os locais de convívio (SAPIEZYNSKI *et al.*, 2015), e principalmente em estabelecimentos de maior porte como os hospitais. Redes Wi-Fi podem ser usadas como modo de entretenimento, informação, mas também podem ser utilizadas como meio de definição da posição dos dispositivos conectados a elas. Esses dispositivos são bastante variados e inseridos atualmente na proposta da Internet das Coisas (IoT), presente principalmente em celulares comuns a vários indivíduos. Um exemplo semelhante de sistema de localização é apresentado pelos Sistemas de Posicionamento Global (GPS) que, a partir da identificação da posição de um dispositivo em relação ao sinal obtido por pelo menos três satélites diferentes, define um ponto comum de localização, minimizando os erros de estimação desses dispositivos. Os dispositivos Wi-Fi funcionam de forma semelhante por ter uma difusão de sinal em um espaço pequeno, onde vários pontos de acesso podem simular o papel dos satélites nesse caso.

Já no caso de códigos QR (QR Codes), é possível armazenar grandes quantidades de informação codificadas em uma imagem, facilmente criada por qualquer pessoa com acesso a um gerador do mesmo. Além disso, esse código pode ser lido por qualquer câmera configurada

para tal. Com isso qualquer dispositivo móvel comum, como os celulares citados anteriormente, ou uma câmera acoplada a um dispositivo embarcado, como uma bandeja “inteligente”, pode difundir e captar essas informações, facilitando o trabalho de um profissional que gerencia diversas pessoas com problemas diferentes no dia-a-dia. O código QR pode ser usado para identificar medicamentos e pessoas (pacientes e médicos) que possuam um código de identificação próprio.

Esse projeto tem então por finalidade unir o método de localização via Wi-Fi com a identificação e armazenamento de dados por QR Codes em um dispositivo embarcado. Para tal, busca-se uma solução barata e eficaz aos problemas na administração hospitalar, verificados nesse projeto.

1.2 Objetivos

O objetivo principal deste trabalho foi desenvolver ferramentas e sistemas computacionais para o apoio a serviços hospitalares. Primeiramente está a construção de um sistema embarcado (bandeja inteligente) que identifique objetos e pessoas, mostrando o impacto que o uso desse sistema pode ter no trabalho de um profissional da área de saúde. Por fim, um método de baixo custo para localizar a própria bandeja e, conseqüentemente, o profissional e paciente que a estiverem utilizando, verificando a possibilidade de implementação desse sistema em um ambiente hospitalar.

Portanto como objetivos mais específicos deste projeto, temos:

- Criação de um sistema computacional embarcado (bandeja inteligente), capaz de processar informações provenientes de uma câmera e de se comunicar com um sistema Wi-Fi;
- Capturar imagens e realizar a leitura de códigos de identificação (QR Codes);
- Desenvolver um método de localização baseado em sinais wireless e passível de uso em ambientes internos (localização Wi-Fi) por meio da técnica de *fingerprinting*.

1.3 Organização da Monografia

O restante deste documento é dividido em 3 capítulos principais. No Capítulo 2 (Métodos, Técnicas e Tecnologias utilizadas) será discutido todos os conceitos e técnicas relevantes para a elaboração do projeto bem como suas referências. Já no Capítulo 3, intitulado Desenvolvimento, será descrita toda a metodologia e a forma como foram integradas e desenvolvidas todas as tecnologias presentes na Bandeja proposta. Por fim, no Capítulo 4 (Conclusão) estão as limitações, dificuldades, conclusões e considerações finais referentes ao trabalho desenvolvido e ao Curso de Graduação.

REVISÃO BIBLIOGRÁFICA

2.1 Considerações Iniciais

Existem muitos artigos e trabalhos que buscam verificar a efetividade da utilização de técnicas de cuidados de saúde (ALLEN; GILLEN; RIXSON, 2009) (ALLEN; RIXSON, 2008), bem como o impacto na utilização de tecnologias para auxiliar o tratamento médico (BLACK *et al.*, 2011). Os resultados obtidos variam por existirem poucas evidências de qualidade para chegar em uma ótima conclusão (ELBERT *et al.*, 2014), porém existem poucos estudos que abordem tecnologias que auxiliem no gerenciamento medicinal em hospitais. Este capítulo será responsável por descrever os conceitos utilizados para elaboração do sistema embarcado, assim como detalhar os artigos que fundamentam e justificam a pesquisa nessa área.

2.2 Conceitos e Técnicas Relevantes

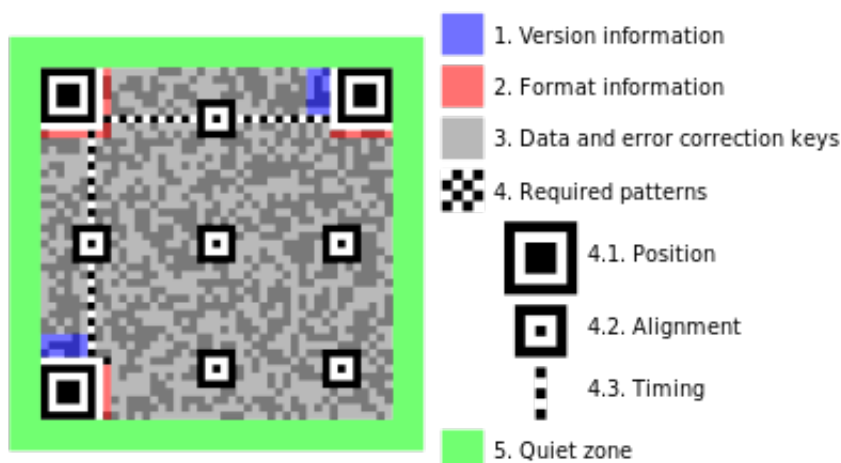
2.2.1 QR Code

O QR Code[Figura 1] é um código matricial de duas dimensões (2D) que permite guardar em um pequeno espaço grandes blocos de dados e sua leitura pode ser feita por qualquer câmera configurada para tal. Seu modo de correção de erros utiliza o código de Reed-Solomon, permitindo a leitura do QR Code mesmo quando sujo ou danificado. Além disso, possui até quarenta versões com volumes de dados diferentes para aplicações diversas (INCORPORATED, 2011).

Existem projetos que utilizaram o QR Code tanto para identificação de remédios (FONG; CHUNG, 2013), tanto na criação de watermarks para autenticação em esquemas de telerradiologia (SEENIVASAGAM; VELUMANI, 2013), assim como registro de dados de pacientes (TSCHANZ; DORNER; DENECKE, 2017). Para a extração dos dados e identificação do QR Code é feita uma triangulação com os quadrados, orientado em proporções 1:1:3:1:1, existentes em três de seus cantos, permitindo a leitura de códigos independente de sua inclinação ou deformação.

O modo de organização e obtenção de dados não é especificado e é gerido por uma ISO, sendo que suas métricas e métodos só são explicados via compra de licença, por isso não serão detalhados nesse trabalho. Sua utilização só foi possível pela existência de bibliotecas com

Figura 1 – Exemplo da estrutura de um QR Code.



Fonte: [Wikipedia](#) (2013).

comandos de leitura de QR Codes, no caso desse trabalho foi utilizado o programa *ZBarcam* criado pelos próprios idealizadores da biblioteca opensource *Zbar* ([BROWN, 2007-2010](#)).

2.2.2 Localização Indoor

Os métodos de localização ([SANTOSO; REDMOND, 2015](#)) em pequenos ambientes internos são os seguintes:

RFID: utiliza sinais de rádio para identificação, podendo recuperar e armazenar dados em tags. São separados entre RFID passivos, que funcionam bem para identificação e necessitam de um terminal de leitura para operar, e ativos, que emite sinais por longas distâncias e são precisos em localização (erro varia de menos de 1 metro até 10 metros). Funcionam muito bem para ambientes fechados por não possuírem muitas barreiras que impedem seu funcionamento, porém são extremamente caros e necessitam de equipamentos próprios para seu funcionamento. Uma abordagem que deriva desse método é a utilização de NFC ([MENG *et al.*, 2014](#)), todavia ainda não existem muitos estudos na área.

GSM: utiliza a informação de antenas e de seu sinal para estimar sua posição. Muito presente em aparelhos celulares, porém com pouca precisão atingindo erros de até 1000 metros, inviável para localização em um ambiente hospitalar.

Bluetooth([MARINELLO *et al.*, 2015](#)): utilização comunicação via redes sem-fio. Possui precisão métrica próxima do Wi-Fi (erro variando de 2 a 10 metros), porém depende da instalação de terminais de acesso Bluetooth que não são tão comuns como roteadores Wi-Fi e são mais sensíveis ao ambiente de instalação que os RFID.

Indoor GPS: utiliza repetidores de sinal de satélite dentro de ambientes fechados para melhorar a identificação já feita pela tecnologia de GPS, ou seja, ampliando o seu alcance e

precisão, de menos de 1 metro de erro. Porém, assim como RFID e Bluetooth, depende da compra de outros equipamentos, tornando-a uma tecnologia cara.

Wi-Fi: utiliza o sinal e identificação de pontos de acesso de redes sem-fio para estimativa da localização do dispositivo (PESSIN *et al.*, 2013) (PESSIN *et al.*, 2014) (PESSIN *et al.*, 2012). Roteadores Wi-Fi estão presentes em vários ambientes e dificilmente um dispositivo não identifica menos de um sinal Wi-Fi (SAPIEZYNSKI *et al.*, 2015), porém é extremamente suscetível a interferências do ambiente apesar do baixo custo e boa precisão (erro variando de 2 a 10 metros). Em contrapartida, pode ser utilizado alternativamente com outros métodos de localização para melhorar sua precisão e diminuir sua interferência.

Métodos conhecidos de estimação da localização por Wi-Fi são: *RSSI*, que mede a força de sinal entre o dispositivo e diversos pontos de acesso criando um modelo de propagação que estima a distância entre eles; *Fingerprinting*, que utiliza a técnica de *RSSI* combinadas com as coordenadas GPS criando uma base de dados utilizada para comparação no momento de estimar a posição do dispositivo, método utilizado nesse trabalho; *AoA*, o Ângulo de Chegada utiliza triangulação do sinal de pontos de acesso para estimar o posicionamento; e o *ToA* ou Tempo de Chegada, que utiliza de timestamps dos sinais dos pontos de acesso para calcular a posição do dispositivo.

2.3 Trabalhos Relacionados

Os trabalhos na área utilizam em sua maioria de tecnologias como o celular pela facilidade de acesso e por ser um dispositivo comum para a maioria das pessoas (NHAVOTO; GRONLUND, 2014). Porém o foco dos cuidados e gerenciamento da dosagem e medicação fica por conta dos pacientes e não do profissional da saúde (MALONE; SAVERNO, 2012) ou a utilização dos sistemas é feito somente para identificação do paciente e sua posição no âmbito global e não em pequenos ambientes como o hospital (LIN *et al.*, 2006). Por isso foi feita a escolha de um sistema embarcado que auxilie diretamente o profissional de saúde, evitando erros no desconhecimento do medicamento prescrito e permitindo um reconhecimento melhor do paciente atendido, além de sua localização, buscando evitar a confusão no momento da medicação.

2.4 Considerações Finais

Nesse capítulo foram descritas as tecnologias empregadas bem como sua utilização na área pesquisada. Foi verificada uma falta de estudos na utilização de métodos que auxiliem a atuação do profissional da saúde na medicação de pacientes em hospitais, atribuindo as soluções do erro geralmente ao paciente, além de poucos casos de mapeamento e localização de pequenos ambientes, utilizando, em sua maioria, tecnologias muito caras para aceitação e inclusão geral, como o RFID ativo.

No próximo capítulo será apresentada a forma como foi elaborado e organizado o sistema da bandeja, bem como seus resultados.

DESENVOLVIMENTO

3.1 Considerações iniciais

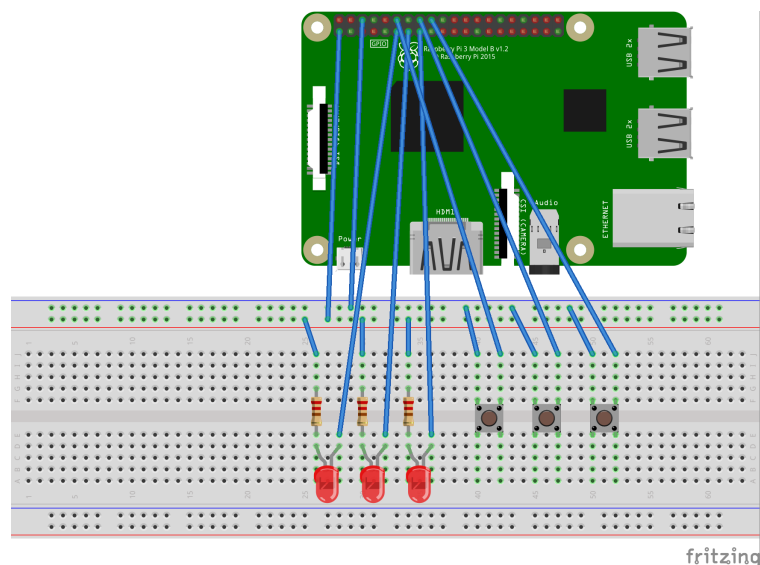
Neste capítulo serão apresentados os métodos utilizados para resolução dos problemas de detecção de QR Code e localização do dispositivo por meio da técnica de *fingerprinting*, bem como foi feita a modularização e junção de ambas as soluções em um só programa gerido por elementos eletrônicos presentes em uma Breadboard acoplada à RaspberryPI.

3.2 Descrição Geral de Funcionamento do Sistema

O sistema funciona por meio de um programa em linguagem Python que gerencia o comportamento de botões e quais ações tomar ao pressioná-los, existem também luzes LEDs como feedback das ações tomadas. Em cada LED foi utilizado um resistor pull-down de 220Ω.

Existem três botões e três rotinas de decisão ilustrados na figura 2.

Figura 2 – Exemplo da organização do circuito da Bandeja



Ao iniciar o programa todos os LEDs são acesos por 3 segundos, demonstrando que o programa está sendo executado. É possível iniciá-lo ao ligar a RaspberryPI incluindo o comando `python ../path-to/StartingTray.py` via `sudo` no arquivo `/etc/rc.local` logo acima do comando `exit 0`.

Nesse caso o feedback é extramente importante para todas as rotinas funcionarem sem problemas. Para iniciar o programa basta digitar *sudo python StartingTray.py*.

Ao pressionar o primeiro botão um script Shell em formato Bash é executado para fazer a leitura do QR Code a partir de uma câmera acoplada à RaspberryPI. Por 10 segundos espera-se que o usuário posicione o código em frente à câmera para ser lido, as informações recuperadas são então salvas em um arquivo de saída e o script retorna como relatório se foi lido com sucesso ou não o código. Um LED de cor vermelha permanece ligado enquanto o sistema processa a requisição.

Já o segundo botão aciona um script Shell, também em formato Bash, que após alguns segundos retorna como relatório a identificação de onde está localizado o dispositivo, se localizado com sucesso, ou falha na identificação, caso ele não tenha sido encontrado. Durante a execução do processo um LED de cor verde fica ligado e desliga ao terminar a requisição.

Por fim o último botão apenas envia um sinal para o sistema desligar, um LED de cor amarela confirma a requisição antes da finalização da execução.

3.3 Descrição das Atividades Realizadas

3.3.1 Sistema Eletrônico

Todas as rotinas de gerenciamento de eventos como o pressionar de um botão e o desligamento/ligamento de um LED foi feita pela biblioteca *RPi.GPIO* (CROSTON, 2016) criada exclusivamente para gerenciar entradas e saídas eletrônicas da RaspberryPI. Essa biblioteca é portada para C++ e Python, no caso todo o projeto foi desenvolvido em Python pela abstração e facilidade de trabalho se comparado a C++. Além disso, foi utilizada a biblioteca *OS*, para interação com o sistema operacional enviando comandos de terminal tanto para executar scripts Shell como para enviar uma requisição para desligar o sistema e a biblioteca *time*, para criar pontos de espera de execução.

Como é possível ver no código de Inicialização do Sistema [Apendice A: Código Fonte 3], a função *main()* executa duas funções por padrão, sendo elas a função *setup()* de pré-configuração dos botões e LEDs e a função *running()* que apenas liga por 3 segundos todos os LEDs e em seguida desliga para informar que o programa está em execução. Além disso, existem outras duas funções: *loop()*, que executa enquanto não houver interrupções (CTRL+C), cria as detecções de eventos ao pressionar os botões e fica em um loop infinito esperando o pressionar de algum deles; já a função *destroy()*, que roda ao ocorrer a interrupção (CTRL+C), desliga todos os LEDs, apaga os eventos e pré-configurações criados no programa, e termina a execução do mesmo.

3.3.2 Sistema de leitura de QR Code

Ao executar a rotina de evento de QR Code é executado o script Shell em Bash *QR-Code.sh*[1].

Código-fonte 1: Script que lê o QR Code

```
1 #!/bin/bash
2 echo "Reading QRCode."
3 zbarcam --nodisplay --raw > QRCodes.txt &
4 PID=$!
5 sleep 5
6 kill $PID
7 if [ -s QRCodes.txt ] ; then
8   echo "QRCode was successfully read."
9 else
10  echo "Problems reading the QRCode."
11 fi
12 #rm -rf QRCodes.txt
```

Na linha 3 do código ele executa em background o programa *zbarcam* de demonstração da biblioteca opensource *Zbar*([BROWN, 2007-2010](#)) para leitura do código QR Code passando as flags *--nodisplay*, que fala para o programa não abrir a interface gráfica e enfrentar problemas de execução por falta de monitor, e *--raw*, para a saída ser somente o código lido, que é então redirecionado para o arquivo *QRCodes.txt*. O PID do processo é salvo na linha 4 e depois de 5 segundos o processo é morto pelo script. Por fim é verificado se o arquivo *QRCodes.txt* existe e se ele não está em branco na linha 7, caso seja verdade é impresso que o código foi lido com sucesso [Figura 7] e impresso que houve problemas na leitura caso contrário [Figura 8].

Foi utilizado um programa de demonstração da biblioteca *Zbar*([BROWN, 2007-2010](#)) pois atendia a proposta do trabalho, além de facilitar a leitura já que não era necessário criar algoritmos para corrigir a inclinação ou deformação do código na exibição do mesmo para a câmera. Essa biblioteca permite a leitura de vários tipos diferentes de códigos de barra, não só todos os padrões de QR Codes, como, por exemplo, EAN-13/UPC-A, UPC-E, EAN-8, Code 128 e Code 39. A leitura é feita por simples métodos como *scan()* (em *Python*), em que todo o pré-processamento (redução de ruído, melhora de contraste, detecção de bordas, por exemplo) da imagem é feito na própria implementação das funções, sem necessidade do usuário codificar essa parte.

3.3.3 Sistema de Localização via Wi-Fi

Ao executar a rotina de evento de Localização é executado o script Shell em Bash *WiFi.sh*[2].

Código-fonte 2: Script que determina a localização da bandeja

```
1 #!/bin/bash
2 echo "Tracking the tray."
3 sudo ./findclient -g USP --nodebug > Localizations.txt
4 if [ -s Localizations.txt ] ; then
5     python ParseLocalizations.py
6     echo "The tray was successfully tracked."
7 else
8     echo "Problems tracking the tray."
9 fi
10 #rm -rf Localizations.txt
```

Para localizar a bandeja foi utilizado o programa open-source *FIND*(SCHOLL, 2015-2016) que utiliza o método de *fingerprinting* para localizar o dispositivo. O programa é dividido entre cliente e servidor, o cliente envia as leituras de sinal Wi-Fi para o servidor e ele retorna com a localização do mesmo. O cadastro de localizações deve ser feito previamente, caso contrário não será encontrado o dispositivo.

Para cadastrar uma localização basta executar o comando "*sudo ./findclient -g YOUR-GROUP -e*", em que a flag *-g* determina o seu grupo de dispositivos e locais cadastrados e a flag *-e* indica que está sendo feito um cadastro de localização, após isso ele pedirá o nome do local, e então os dados das leituras de sinal Wi-Fi serão enviados para o servidor existente do *FIND*(SCHOLL, 2015-2016). É possível também criar o seu próprio servidor local seguindo as instruções do site, porém não foi utilizado nesse trabalho essa opção.

O servidor permite a utilização de três redes neurais diferentes para treinar e classificar a localização do dispositivo: **SVM**: utiliza uma formulação matemática para criar a divisão entre duas classes em um espaço multidimensional; **Naive-Bayes**: usa o teorema probabilístico de Naive-Bayes para calcular a probabilidade de features independentes, esse é o modelo padrão do servidor e o que está sendo utilizado neste trabalho; **Random Forest**: calcula a probabilidade de ocorrência de todas as features de forma independente e depois junta a probabilidade de ocorrência do caso mais próximo de todas as features do dado que está sendo classificado, para melhorar sua acurácia.

Para localizar um dispositivo basta executar o comando "*sudo ./findclient -g YOUR-GROUP*". Como exemplo, na linha 3 do código 2, está sendo usado o grupo USP, criado para esse trabalho, junto com a flag *--nodebug* para a saída ser somente os dados da leitura de localização. O servidor lê três vezes o sinal Wi-Fi e retorna como saída o resultado de todas as leituras, que é então redirecionado para o arquivo *Localizations.txt*.

Após a execução do processo, é verificado então se existe o arquivo *Localizations.txt* e se ele não está vazio. Caso seja verdade: executa-se o programa *ParseLocalizations.py*[Apendice A: Código Fonte 4] que verifica entre as três leituras feitas qual foi o resultado e retorna a localização

aproximada conforme o resultado, é então impresso que a bandeja foi encontrada com sucesso [Figura 6]. Caso contrário: é impresso que houve problemas para localizar a bandeja.

3.4 Resultados Obtidos

O objetivo inicial de desenvolver um sistema, focado no baixo custo, que consiga identificar medicamentos, pacientes e profissionais da saúde por meio de QR Codes, e possua um sistema de localização em pequenos ambientes via medida de sinal Wi-Fi foi alcançado.

Para validar a localização foram enviados 100 leituras de sinal Wi-Fi de cada ambiente cadastrado para melhorar sua detecção, é possível ver precisão de acerto de cada registro na Figura 3. A rede utilizou entre 73 a 74 de cada uma dessas leituras para ser treinada e obteve uma acurácia próxima de 100% em todos os casos. Os testes foram feitos em uma residência, porém os resultados devem ser próximos em ambientes maiores como a USP, já que existe um maior volume de roteadores Wi-Fi identificados.

Figura 3 – Dados da leitura do sinal Wi-Fi

1 (Mixin = 0.7, Cutoff = 0.005)		
home suite restroom	(74)	100%
home bedroom	(74)	100%
home kitchen	(74)	100%
home dinning room	(73)	100%
home office	(74)	100%
home restroom	(73)	98%

Já no caso da detecção do QR-Code é necessária uma distância mínima para operar sem problemas, além de uma boa iluminação no caso de câmeras com baixa qualidade de imagem. Com a câmera utilizada foi observado um bom funcionamento com pelo menos 20 centímetros de distância da câmera no caso com boa iluminação, já no caso de pouca iluminação a distância necessária dobra para 40 cm como é possível observar no vídeo [Apendice B: Testes e material utilizado B].

Em comparação a outros estudos da área, este trabalho utiliza de técnicas e algoritmos estudados em separado, com poucas evidências semelhantes à junção proposta e alcançada, porém com deficiências já explicitadas por estudos anteriores ainda que cumpra seu propósito inicial.

3.5 Dificuldades, Limitações e Trabalhos Futuros

Um limitante desse projeto foi a falta de acessibilidade ao modo de leitura dos dados dos QR Codes. Inicialmente a ideia era utilizar a biblioteca *OpenCV* para fazer a identificação do

código e extração dos dados, porém não foi possível por essa limitação, sendo necessária a busca de alternativas como o *ZBar*.

Os tópicos que o sistema enfrenta problemas são:

1. A qualidade e interferência de sinal Wi-Fi presente pela organização de infraestrutura de cada prédio bem como a movimentação de equipamentos e objetos metálicos de forma constante nos hospitais (ISSOM *et al.*, 2012);
2. O acesso e o preço de produtos semelhantes à RaspberryPI no Brasil, já que não existe, até o momento dessa pesquisa, produção nacional desses recursos e as que existem são de preço um tanto elevado (devido principalmente aos custos de importação) elevado, saindo da proposta desse trabalho.

Uma boa abordagem prática e acessível, tanto ao profissional de saúde como também ao paciente, seria a conexão desse dispositivo com um aparelho celular por meio da criação de um aplicativo próprio, tornando a localização e a interação com o sistema mais fluída e menos robusta. Desse modo também seria possível expandir as funcionalidades do sistema criando uma base de dados acessível ao paciente, para lembrar o mesmo de se medicar através de notificações, e ao médico ou enfermeiro, agilizando a consulta do registro hospitalar.

3.6 Considerações Finais

Nesse capítulo foram descritos a forma de organização do sistema embarcado tanto no âmbito eletrônico como em sua estrutura de código, descrevendo como foram utilizadas as bibliotecas *RPi.GPIO* e *ZBar*, a interação dos programas em Python com scripts Shell em Bash, assim como o funcionamento de uma forma geral do programa opensource *FIND* e suas configurações de utilização. Foram também discutidos os resultados obtidos, concluídos de modo satisfatório, assim como as dificuldades e limitações do sistema proposto.

O próximo capítulo traz as conclusões referentes aos estudos.

CONCLUSÃO

4.1 Considerações Sobre o Projeto Desenvolvido

Ainda existe, de modo geral, uma necessidade por soluções que evitem erro na compra, prescrição e aplicação de medicações. O objetivo do projeto foi alcançado, mostrando que é sim possível a elaboração de um dispositivo de baixo custo que auxilie hospitais no gerenciamento de pacientes, medicamentos e recursos nos métodos descritos e com a utilização de mini-computadores de baixo orçamento como a RaspberryPI.

Esse estudo permitiu o aprendizado de técnicas de localização indoor, o funcionamento e utilização de identificadores tanto 2D como a rádio frequência, além de expandir o conhecimento no desenvolvimento eletrônico em minicomputadores e novas abordagens na área de Aprendizado de Máquina.

4.2 Considerações sobre o Curso de Graduação

O curso de graduação em Ciências de Computação oferecido pelo Instituto de Ciências Matemáticas e de Computação (ICMC) aborda em sua grade curricular temas importantes e necessários para formar um profissional bem preparado, com conhecimento sobre diversos campos na área e pensamento crítico e científico em abordagens tanto teóricas como práticas.

Porém faltam incorporar aspectos como:

1. Reorganização da grade curricular, pois muitos temas importantes para o conhecimento e o contato com certas áreas, que possam interessar ao aluno, são feitas tardiamente;
2. Créditos de trabalho deveriam ser melhor aproveitados e contabilizados, pois, apesar de estarem presentes e serem importantes no desenvolvimento, não são bem valorizados;
3. Estimular mais a participação em atividades extracurriculares, seja com maior divulgação dos grupos e laboratórios ou também com a criação de matérias no formato das já oferecidas Atividades Acadêmicas Científicas, de Extensão e Culturais I e II.

REFERÊNCIAS

ALLEN, D.; GILLEN, E.; RIXSON, L. The effectiveness of integrated care pathways for adults and children in health care settings: A systematic review. **JBIC Libr Syst Rev**, v. 7, n. 3, p. 80–129, 2009. ISSN 1838-2142. Citado na página 23.

ALLEN, D.; RIXSON, L. How has the impact of 'care pathway technologies' on service integration in stroke care been measured and what is the strength of the evidence to support their effectiveness in this respect? **Int J Evid Based Healthc**, v. 6, n. 1, p. 78–110, 2008. ISSN 1744-1595 (Print) 1744-1595. Citado na página 23.

BLACK, A. D.; CAR, J.; PAGLIARI, C.; ANANDAN, C.; CRESSWELL, K.; BOKUN, T.; MCKINSTRY, B.; PROCTER, R.; MAJEED, A.; SHEIKH, A. The impact of ehealth on the quality and safety of health care: a systematic overview. **PLoS Med**, v. 8, n. 1, p. e1000387, 2011. ISSN 1549-1277. Citado na página 23.

BROWN, J. **ZBar bar code reader**. 2007–2010. Accessed: 2017-06-09. Disponível em: <<http://zbar.sourceforge.net/>>. Citado 2 vezes nas páginas 24 e 29.

CROSTON, B. **RPi.GPIO Python Module**. 2016. Accessed: 2017-06-09. Disponível em: <<https://sourceforge.net/p/raspberry-gpio-python/wiki/Home/>>. Citado na página 28.

ELBERT, N. J.; OS-MEDENDORP, H. van; RENNELAAR, W. van; EKELAND, A. G.; ROIJEN, L. Hakkaart-van; RAAT, H.; NIJSTEN, T. E.; PASMANS, S. G. Effectiveness and cost-effectiveness of ehealth interventions in somatic diseases: a systematic review of systematic reviews and meta-analyses. **J Med Internet Res**, v. 16, n. 4, p. e110, 2014. ISSN 1438-8871. Citado na página 23.

FONG, E. M.; CHUNG, W. Y. Mobile cloud-computing-based healthcare service by noncontact ecg monitoring. **Sensors (Basel)**, v. 13, n. 12, p. 16451–73, 2013. ISSN 1424-8220. Citado na página 23.

INCORPORATED, D. W. **Denso QR Code Essentials**. 2011. Accessed: 2017-06-09. Disponível em: <<https://www.nacs.org/LinkClick.aspx?fileticket=D1FpVAvvJuo%3D&tabid=1426&mid=4802>>. Citado na página 23.

ISSOM, D. Z.; HAGRY, C.; MENDO, L. W.; SENG, H.; EHRLER, F.; LOVIS, C. Challenges and issues of geolocation in clinical environment. **Stud Health Technol Inform**, v. 180, p. 447–51, 2012. ISSN 0926-9630 (Print) 0926-9630. Citado na página 32.

LIN, C. C.; CHIU, M. J.; HSIAO, C. C.; LEE, R. G.; TSAI, Y. S. Wireless health care service system for elderly with dementia. **IEEE Trans Inf Technol Biomed**, v. 10, n. 4, p. 696–704, 2006. ISSN 1089-7771 (Print) 1089-7771. Citado 2 vezes nas páginas 21 e 25.

MALONE, D. C.; SAVERNO, K. R. Evaluation of a wireless handheld medication management device in the prevention of drug-drug interactions in a medicaid population. **J Manag Care Pharm**, v. 18, n. 1, p. 33–45, 2012. ISSN 1083-4087. Citado 2 vezes nas páginas 21 e 25.

- MARINELLO, S. H.; UEYAMA, J.; PESSIN, G.; OSÓRIO, F. S.; VARGAS, P. A. Determinando posições relativas de nós móveis por meio de bluetooth e aprendizado de máquina. **Revista de Exatas e TECnológicas**, v. 1, n. 3, 2015. ISSN 2236-739X. Citado na página 24.
- MENG, P.; FEHRE, K.; RAPPELSBERGER, A.; ADLASSNIG, K. P. Framework for near-field-communication-based geo-localization and personalization for android-based smartphones—application in hospital environments. **Stud Health Technol Inform**, v. 198, p. 9–16, 2014. ISSN 0926-9630 (Print) 0926-9630. Citado na página 24.
- NHAVOTO, J. A.; GRONLUND, A. Mobile technologies and geographic information systems to improve health care systems: a literature review. **JMIR Mhealth Uhealth**, v. 2, n. 2, p. e21, 2014. ISSN 2291-5222. Citado na página 25.
- PESSIN, G.; OSÓRIO, F. S.; SOUZA, J. R.; COSTA, F. G.; UEYAMA, J.; WOLF, D. F.; BRAUN, T.; VARGAS, P. A. Evolving an indoor robotic localization system based on wireless networks. In: **International Conference on Engineering Applications of Neural Networks**. [S.l.]: Springer, 2012. p. 61–70. ISBN 364232908X. Citado na página 25.
- PESSIN, G.; OSÓRIO, F. S.; SOUZA, J. R.; UEYAMA, J.; COSTA, F. G.; WOLF, D. F.; DIMITROVA, D.; BRAUN, T.; VARGAS, P. A. Investigation on the evolution of an indoor robotic localization system based on wireless networks. **Applied Artificial Intelligence**, v. 27, n. 8, p. 743–758, 2013. ISSN 0883-9514. Citado na página 25.
- PESSIN, G.; OSÓRIO, F. S.; UEYAMA, J.; WOLF, D. F.; MOIOLI, R. C.; VARGAS, P. A. Self-localisation in indoor environments combining learning and evolution with wireless networks. In: **Proceedings of the 29th Annual ACM Symposium on Applied Computing**. [S.l.]: ACM, 2014. p. 661–666. ISBN 145032469X. Citado na página 25.
- SANTOSO, F.; REDMOND, S. J. Indoor location-aware medical systems for smart homecare and telehealth monitoring: state-of-the-art. **Physiol Meas**, v. 36, n. 10, p. R53–87, 2015. ISSN 0967-3334. Citado na página 24.
- SAPIEZYNSKI, P.; STOPCZYNSKI, A.; GATEJ, R.; LEHMANN, S. Tracking human mobility using wifi signals. **PloS one**, v. 10, n. 7, p. e0130824, 2015. ISSN 1932-6203. Citado 2 vezes nas páginas 21 e 25.
- SCHOLL, Z. **Framework for Internal Navigation and Discovery**. 2015–2016. Accessed: 2017-06-09. Disponível em: <<https://www.internalpositioning.com/>>. Citado na página 30.
- SEENIVASAGAM, V.; VELUMANI, R. A qr code based zero-watermarking scheme for authentication of medical images in teleradiology cloud. **Comput Math Methods Med**, v. 2013, p. 516465, 2013. ISSN 1748-670x. Citado na página 23.
- TSCHANZ, M.; DORNER, T. L.; DENECKE, K. emedication meets ehealth with the electronic medication management assistant (emma). In: **Health Informatics Meets EHealth: Digital Insight—Information-Driven Health & Care. Proceedings of the 11th EHealth2017 Conference**. [S.l.]: IOS Press, 2017. v. 236, p. 196. ISBN 1614997594. Citado na página 23.
- WIKIPEDIA. **QR Code Structure**. 2013. Accessed: 2017-06-09. Disponível em: <https://en.wikipedia.org/wiki/QR_code>. Citado na página 24.

APÊNDICE A

CÓDIGOS UTILIZADOS

Código-fonte 3: Programa que gerencia o sistema

```
1 import RPi.GPIO as GPIO
2 import time
3 import os
4
5 FstBtnPin = 12
6 SndBtnPin = 16
7 TrdBtnPin = 18
8
9 RedLedPin = 11
10 GreenLedPin = 13
11 YellowLedPin = 15
12
13 RedLed_status = 1
14 GreenLed_status = 1
15 YellowLed_status = 1
16
17 def setup():
18     GPIO.setmode(GPIO.BOARD)
19     GPIO.setup(RedLedPin, GPIO.OUT)
20     GPIO.setup(GreenLedPin, GPIO.OUT)
21     GPIO.setup(YellowLedPin, GPIO.OUT)
22     GPIO.setup(FstBtnPin, GPIO.IN, pull_up_down=GPIO.PUD_UP)
23     GPIO.setup(SndBtnPin, GPIO.IN, pull_up_down=GPIO.PUD_UP)
24     GPIO.setup(TrdBtnPin, GPIO.IN, pull_up_down=GPIO.PUD_UP)
25     GPIO.output(RedLedPin, GPIO.HIGH)
26     GPIO.output(GreenLedPin, GPIO.HIGH)
27     GPIO.output(YellowLedPin, GPIO.HIGH)
28
29 def scanQRCode(ev=None):
30     global RedLed_status
31     RedLed_status = not RedLed_status
32     GPIO.output(RedLedPin, RedLed_status)
33     os.system("/home/pi/Documents/TCC/QRCode.sh")
34     RedLed_status = not RedLed_status
35     GPIO.output(RedLedPin, RedLed_status)
36
```

```
37 def findMedicalTray(ev=None):
38     global GreenLed_status
39     GreenLed_status = not GreenLed_status
40     GPIO.output(GreenLedPin, GreenLed_status)
41     os.system("/home/pi/Documents/TCC/WiFi.sh")
42     GreenLed_status = not GreenLed_status
43     GPIO.output(GreenLedPin, GreenLed_status)
44
45 def shutdownMedicalTray(ev=None):
46     global YellowLed_status
47     YellowLed_status = not YellowLed_status
48     GPIO.output(YellowLedPin, YellowLed_status)
49     print("The system will be shutdown. Thank you for using this
        application.")
50     time.sleep(0.5)
51     destroy()
52     os.system("sudo poweroff")
53
54 def loop():
55     GPIO.add_event_detect(FstBtnPin, GPIO.FALLING, callback=scanQRCode,
        bouncetime=200)
56     GPIO.add_event_detect(SndBtnPin, GPIO.FALLING, callback=
        findMedicalTray, bouncetime=200)
57     GPIO.add_event_detect(TrdBtnPin, GPIO.FALLING, callback=
        shutdownMedicalTray, bouncetime=200)
58     while True:
59         time.sleep(1)
60
61 def destroy():
62     GPIO.output(RedLedPin, GPIO.HIGH)
63     GPIO.output(GreenLedPin, GPIO.HIGH)
64     GPIO.output(YellowLedPin, GPIO.HIGH)
65     GPIO.cleanup()
66
67 def running():
68     global RedLed_status
69     RedLed_status = not RedLed_status
70     GPIO.output(RedLedPin, RedLed_status)
71     global GreenLed_status
72     GreenLed_status = not GreenLed_status
73     GPIO.output(GreenLedPin, GreenLed_status)
74     global YellowLed_status
75     YellowLed_status = not YellowLed_status
76     GPIO.output(YellowLedPin, YellowLed_status)
77     time.sleep(3)
78     RedLed_status = not RedLed_status
79     GPIO.output(RedLedPin, RedLed_status)
```

```
80  GreenLed_status = not GreenLed_status
81  GPIO.output(GreenLedPin, GreenLed_status)
82  YellowLed_status = not YellowLed_status
83  GPIO.output(YellowLedPin, YellowLed_status)
84
85  if __name__ == "__main__":
86      setup()
87      running()
88      try:
89          loop()
90      except KeyboardInterrupt:
91          destroy()
```

Código-fonte 4: Programa que determina a localização segundo as leituras do FIND

```
1  file = open("Localizations.txt", "r")
2  text = file.readlines()
3  file.close()
4
5  FstLocal = text[0][13:]
6  SndLocal = text[1][13:]
7  TrdLocal = text[2][13:]
8  FinalLocal = ""
9
10 if (FstLocal == SndLocal):
11     if (FstLocal == TrdLocal):
12         FinalLocal = "It's in " + FstLocal
13     else:
14         FinalLocal = "Somewhere near " + FstLocal + " and " + TrdLocal +
            "."
15 else:
16     if (FstLocal == TrdLocal):
17         FinalLocal = "Somewhere near " + FstLocal + " and " + SndLocal +
            "."
18     else:
19         FinalLocal = "We can't determinate exactly where it was suppose
            to be."
20
21 file = open("Localization.txt", "w")
22 file.write(FinalLocal)
23 file.close()
```

APÊNDICE B

TESTES E MATERIAL UTILIZADO

Figura 4 – Foto da Bandeja

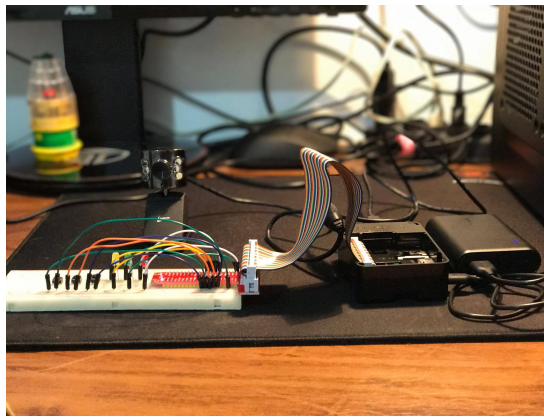


Figura 5 – Foto do circuito da Bandeja

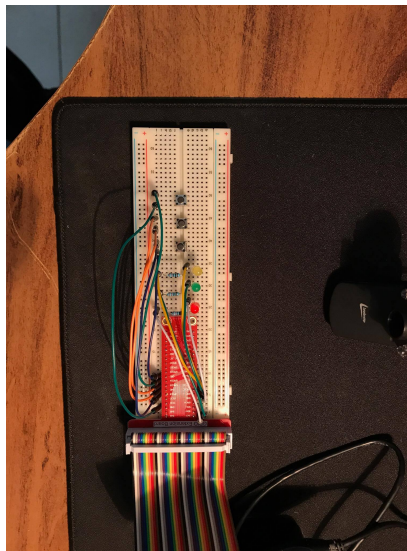


Figura 6 – Exemplo da execução correta da localização Wi-Fi

```
pi@raspberrypi:~/Documents/ICC $ sudo python StartingTray.py
Tracking the tray.
Traceback (most recent call last):
  File "StartingTray.py", line 1, in <module>
    cli_app = cli.App()
  File "/usr/local/lib/python3.7/dist-packages/cli/app.py", line 10, in __init__
    self.action_signature = cli.ActionFunc
AttributeError: 'cli.App' object has no attribute 'action_signature'
The tray was successfully tracked.
```

<<https://www.youtube.com/watch?v=aUxwQXraoEk>> : Vídeo mostrando a execução de todos os módulos da Bandeja Inteligente;

Figura 7 – Exemplo da execução correta da leitura do QR Code

```
*@aspierpp1:~/Documents/ICC $ sudo python StartingRay.py
Reading QRCode.
QRCode was successfully read.
```

Figura 8 – Exemplo da execução falha da leitura do QR Code

```
*@aspierpp1:~/Documents/ICC $ sudo python StartingRay.py
Reading QRCode.
Problems reading the QRCode.
Reading QRCode.
Problems reading the QRCode.
```

[<https://goo.gl/RTN3us>](https://goo.gl/RTN3us) : QR Codes gerados e utilizados para validação do leitor de códigos.

Os dados de cada código possuem a seguinte estrutura: "USPQR", para identificar que o código é feito pela USP, "00" dois dígitos para identificar se é um código de remédio, paciente ou profissional da saúde, "0000000000" dez dígitos de identificação própria, e "X" para validação.

[<http://goqr.me/>](http://goqr.me/) : Site utilizado para geração dos QR Codes

DOCUMENTAÇÃO DE BIBLIOTECAS

<<https://sourceforge.net/p/raspberry-gpio-python/wiki/Examples/>> : Página em inglês com a documentação da biblioteca *RPi.GPIO*;

<http://zbar.sourceforge.net/api/zbar_8h.html> : Página em inglês com a documentação da biblioteca *ZBar*;

<<https://docs.python.org/2/library/os.html>> : Página em inglês com a documentação da biblioteca *OS*;