JOSÉ HENRIQUE DE JESUS

ESTUDO DE TECNOLOGIAS IOT PARA RECONHECIMENTO DE IMAGENS QRCODE OTIMIZADO NO CONTROLE DE SAÍDA DE PRODUTOS DE ALMOXARIFADO

Assis/SP
2018

JOSÉ HENRIQUE DE JESUS

ESTUDO DE TECNOLOGIAS IOT PARA RECONHECIMENTO DE IMAGENS QRCODE OTIMIZADO NO CONTROLE DE SAÍDA DE PRODUTOS DE ALMOXARIFADO

Trabalho de conclusão de Cursos apresentado ao curso de Bacharelado em Ciências da Computação do Instituto Municipal de Ensino Superior de Assis – IMESA e a Fundação Educacional do Município de Assis – FEMA, como requisito à obtenção do Certificado de Conclusão.

Orientando: José Henrique de Jesus
Orientador: Prof. Douglas Sanches da Cunha

Assis/SP
2018

FICHA CATALOGRÁFICA

JESUS, José Henrique.
ESTUDO DE TECNOLOGIAS IOT PARA RECONHECIMENTO DE IMAGENS QRCODE OTIMIZADO NO CONTROLE DE SAÍDA DE PRODUTOS DE ALMOXARIFADO. José Henrique de Jesus. Fundação Educacional do Município de Assis –FEMA – Assis, 2018.
39p.

1. Raspberry Pi 2. QR Code. 3. IOT

CDD:005.304
Biblioteca da FEMA

JOSÉ HENRIQUE DE JESUS

Trabalho de Conclusão de Curso apresentado ao Instituto Municipal de Ensino Superior de Assis, como requisito do Curso de Graduação, avaliado pela seguinte comissão examinadora:

Orientador: _____
Douglas Sanches da Cunha

Examinador: _____
Me. Fernando Cesar de Lima

Assis/SP
2018

DEDICATÓRIA

Dedico este trabalho a meus amigos que me deram suporte, familiares que me ajudaram, a meu orientador pela ajuda e incentivo presente no projeto e a todos os professores que me ajudaram nessa caminhada ao decorrer do curso.

“Não penso muito em legado para as próximas gerações. Penso apenas em acordar de manhã e trabalhar com pessoas brilhantes para criar coisas que, espero, sejam tão apreciadas por outras pessoas como são apreciadas por nós.”

Steves Jobs

“Fracasso é uma possibilidade por aqui. Se as coisas não estão fracassando, você não está inovando o suficiente.”

Elon Musk

RESUMO

Este projeto tem como objetivo analisar em um meio prático uma tecnologia que está cada vez mais presente no nosso dia a dia, que são os QR Codes.
Para isso é utilizado um componente com base no conceito da arquitetura de IOT chamado *Raspberry Pi* que oferece grandes possibilidades de desenvolvimento de projetos variados a um preço acessível.
Para complementar, nesse projeto o dispositivo tem o auxílio de componentes que tornam seu manuseio simplificado e muito abrangente, como: uma câmera que realizará a leitura de QR Codes impressos em pequenas etiquetas que são analisados em um software de reconhecimento e armazena esses dados em arquivos simplificados como arquivos de Texto ou CSV para serem manipulados posteriormente.

Palavras-chave: IOT, Raspberry Pi, QRCode.

ABSTRACT

This project aims to analyze in a practical means a technology that is increasingly present day by day, which are the QR Codes.
Then, we used a component based on the IOT architecture concept called Raspberry Pi that offers great possibilities of development in several projects with an affordable price.
Therefore, in this project the device has the aid of components that become its simplified handling and very comprehensive, such as: a camera that will perform of the QR Codes reading printed on small labels that are analyzed in a recognition software and stores this data in files simplified as text files or CSV to be handled later..

Keywords: IOT, Raspberry Pi, QRCode.

Lista de Ilustrações

SUMÁRIO

1. INTRODUÇÃO

Com o avanço da tecnologia em várias áreas, principalmente que trata de dispositivos conectados a internet, também conhecido como o conceito de *Internet of Things* (Internet das Coisas), a procura por meios que facilitem o acesso ou o desempenho do cotidiano ou até mesmo no interior de empresas é muito procurado.

Empresas procuram cada vez mais, formas de aumentar seu rendimento e evitar perdas desnecessárias de tempo e dinheiro.

Um dos pontos onde ocorre uma certa vazão que se transforma em foco de preocupações é o caso de áreas onde ocorre grande movimentação de objetos como por exemplo o Almoxarifado ou estoques de matéria prima.

Visto esse problema, surgem algumas formas de tentar realizar esse controle de forma a reduzir a taxa de vasão e até evitar gastos desnecessários. (GONÇALVES,2016)

Dessa forma surgiu a ideia de utilizar um componente chamado *Raspberry PI*, que é um microcomputador do tamanho de um cartão de crédito que possibilita através de sua programação, realizar várias tarefas que um computador de grande porte executa.

Como o *Raspberry PI* permite que conecte outros tipos de dispositivos como, telas de *LCD, sensores* ou *uma câmera para captar imagens,* é possível a criação de um dispositivo que possa auxiliar o controle de estoque desses locais de grande movimentação de mercadoria (RASPBERRY,2018).

O objetivo desse projeto tem a ideia, de trazer comodidade e confiabilidade aos funcionários do setor de estoque através da coleta do número dos itens que são manipulados, reduzindo a vazão de produtos sem o conhecimento dos mesmos e assim evitando transtorno como, o de conferências desnecessárias para determinar se todos os itens estão em seus devidos lugares.

Para realizar isso, os usuários contarão com um dispositivo que realizara a leitura dos códigos QR(*Código de resposta rápida "Quick Response"*) anexados a cada item através d*a câmera* e armazenará esses dados coletados a fim de realizar a baixa no estoque final.

Atualmente no ambiente que será utilizado para teste, a retirada de um item do almoxarifado é anotada em uma folha contendo o código, quantidade e setor no qual será utilizado.

Infelizmente em alguns casos, essa anotação não é feita, gerando uma necessidade de conferência de todo o estoque em uma frequência semanal.

Com o auxílio do dispositivo, a retirada de um item só será feita após alguns requisitos serem atendidos com base em regras que o sistema irá impor:

- Ser escaneado o código de barras que será anexado ao produto,
- Informar a quantidade de itens que será retirado e o setor no qual se destinará o produto

Esses dados serão armazenados no sistema e depois processados no banco de dados.

Como a conferência de estoque onde há uma elevada quantia de itens leva uma grande parcela de tempo, tentar evitar pode gerar uma maior comodidade aos funcionários responsáveis e também gera maior controle sobre a necessidade de adquirir algum item para suprir o estoque pois o sistema informará com certa precisão a quantidade exata em estoque sem a necessidade de uma ação humana de conferência.

Como o projeto trabalha em torno do *Raspberry Pi,* que é um dispositivo ainda novo em estudo, isso pode fornecer dados relevantes para outras pessoas que possam ter ideias semelhantes de desenvolvimento e que podem assim, gerar soluções para outras áreas com base nos conceitos de *IOT.*

Para o desenvolvimento do projeto, será utilizado *Raspberry Pi 3* como base, uma tela de LCD para o visual e interação e uma câmera para leitura dos códigos QR

Todos os dados que serão coletados são tratados por um *software* desenvolvido utilizando a linguagem *Python,* que armazenará essas informações em um arquivo CSV(Arquivo com valores separados por vírgula) temporário.

Este trabalho será apresentado em 7 capítulos:

- **Introdução;**
- **Capítulo 2 – Conceito de IOT:** Descreve como podemos utilizar a técnologia a nosso favor e gerar novas idéias para pequenas e grandes empresas
- **Capítulo 3 – Raspberry Pi 3:** Demonstra as funcionalidades e especificações do Raspberry Pi levantando formas de trabalhar com o equipamento de forma variada.
- **Capítulo 4: QR Code:** É apresentado um estudo detalhado da forma como o QR Code é construído e como ele organiza os diversos tipos de dados que podem ser armazenados em seu conteúdo.
- **Capítulo 5 – Linguagem Python:** Apresenta as principais características da linguagem e traz alguns exemplos de bibliotecas que podem ser aplicadas quanto ao estudo relacionado a IOT.
- **Capítulo 6 – Decodificador QR:** Apresentação prática dos conceitos estudados durante o processo de aprendizagem utilizando um exemplo de coleta de informações através de análise de QR Code.
- **Conclusão;**

2. CONCEITO DE IOT

O termo Internet das Coisas ou IOT (*Internet os Things)*, surgiu por volta de 1999 e foi criada pelos pesquisadores britânicos, Kevin Ashton, do Instituto de Tecnologia de Massachusetts, com a ideia inicial de somente etiquetar eletronicamente os produtos em uma linha de produção de empresas, facilitando a logística, a partir de identificadores de radiofrequência.

Figura 1: Ilustração conceito IOT

[Local: http://energiainteligenteufjf.com/como-funciona/como-funciona-internet-das-coisas/]

Com o passar do tempo essa visão se expandiu de modo que percebeu-se a oportunidade de adicionar esse conceito em dispositivos de várias formas, tamanhos e fins.

A IOT é vista com bons olhos quando o assunto é inovação. Atualmente vimos muitas formas de como essa tecnologia pode e é utilizada, seja em sustentabilidade, como por exemplo, sensores no sistema elétrico que gerencia a alimentação de eletricidade por toda uma casa e disponibiliza o controle de tudo a partir de um smartfone.

Essa tecnologia ao olhos de (Atzori et al. 2010) e (Gubbi et al.2013) é descrita como um mundo repleto de dispositivos e objetos que fazem parte do cotidiano das pessoas e que podem passar por despercebido, tais objetos que são capazes de interagir com o ambiente, monitorar processos, coletar dados e analisá-los, trocar informações, obedecer coman-

dos e executar informações de forma coordenada e proativa para atender as necessidades do usuário.

Como (Atzori et al. 2010) identificou, o paradigma da Internet das Coisas pode ser classificado em três diferentes visões: orientado a “coisas”(sensores), orientada a Internet(*middleware*) ou orientada a semântica (conhecimento).

Todos os dias surgem novas ideias de como aplicar tecnologia a dispositivos em nossa volta e até podemos vestir eles, graças a tecnologia de *Wearables.*

E como se trata de uma tecnologia inovadora ela pode ser utilizada em vários ambientes, seja no transporte, medicina, cidades inteligentes e até em empresas.

Para as empresas por exemplo, temos várias formas de aplicar esses conceitos a fim de melhorar como os processos são realizados sejam os relacionados a clientes, fornecedores, sistemática de vendas, marketing, operações ou controle de logística.

Levando como exemplo o controle de logística temos como exemplo o uso de sensores RFID utilizados para o controle de malas em aeroportos em uma tentativa de reduzir ao máximo o problema de perca de malas dos usuários, pois quando é notado que uma dessas malas estão fora da rota que deveria seguir, é emitido um alerta para que ela seja encaminhada novamente para o destino correto.(TECNOBLOG).

Outro assunto que atualmente está em destaque são as chamadas Industrias 4.0 que reúnem automação, Internet das Coisas, Big Data, Computação nas nuvens e outras formas inteligentes de lidar com problemas presentes em grandes ou pequenas empresas.

Segue imagem ilustrando um modelo de empresa baseada na arquitetura 4.0

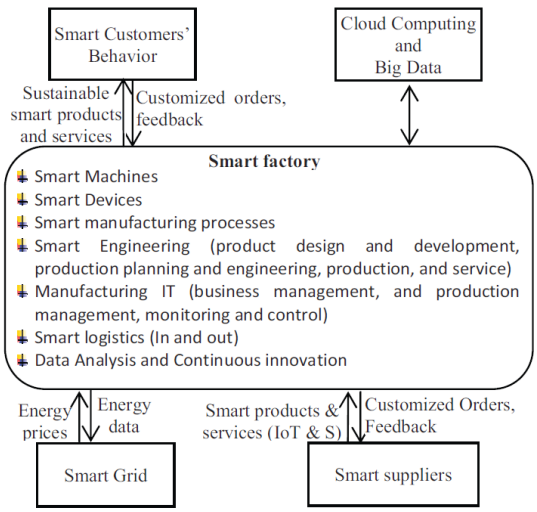

Figura 2: Arquitetura para empresas baseada em IOT

[Local:http://www.decom.ufop.br/imobilis/industria-4-0-e-iot/]

Nesse tipo de arquitetura leva-se em consideração os seguintes tópicos:

- Máquinas Inteligentes: comunicação entre máquinas e interação com o ser humano
- Dispositivos inteligentes: dispositivos conectados a fábrica, tais como sensores, dispositivos móveis, etc.
- Processos inteligentes: Processos automatizados e em tempo real para a gestão e controle de um ambiente de produção ativado por IoT.
- Engenharia Inteligente: inclui o desenvolvimento de produtos, engenharia de produto, produção e serviço pós-venda.
- Fabricação: aplicativos de software utilizados por uma ou mais empresas, monitoramento inteligente e controle por meio de sensores, medidores inteligentes e dispositivos móveis inteligentes
- Logística: ferramentas e processos logísticos inteligentes que reagem a mudanças inesperadas na produção, tais como os gargalo.
- Big Data e Cloud computing: algoritmos de análise que traz grandes oportunidades para melhorar futuras fábricas e processos de fabricação
- Fornecedores inteligentes: construção de relações duradouras com os fornecedores e da mesma forma, aumentar a flexibilidade ao selecionar o melhor fornecedor.
- Rede inteligente: fábrica inteligente no campo de fornecimento de energia para reagir às mudanças nos preços de energia

3. RASPBERRY PI 3

Para conciliar o uso de componentes eletrônicos e análise de imagens, foi preciso encontrar um equipamento que suportasse o necessário se tratando de poder de Hardware. Para isso o componente escolhido foi o Raspberry Pi 3 por sua facilidade de uso e sua capacidade de trabalhar com sensores e componentes diversos. O Raspberry Pi é um componente do tamanho de um cartão de crédito, de baixo custo que possui suporte para vários dispositivos e está sendo muito utilizado para projetos que envolvem IOT. Possibilitando uma nova forma de interação com o mundo real, ele pode ser muito maleável para o que se deseja projetar.(RASPBERRY.ORG)

Nesse projeto será utilizado o Modelo Pi 3, a imagem a seguir demonstra sua representação visual.

Figura 3: Raspberry Pi 3 Modelo B

[Local: https://raspi.tv/2016/raspberry-pi-3-model-b-launches-today-64-bit-quad-a53-1-2-ghz-bcm2837]

Mesmo sendo um dispositivo aparentemente pequeno, ele conta com especificações que conseguem atender a muitos tipos de projetos.

Ele possui em sua composição:

- Processador Quad Core 1.2GHz Broadcom BCM2837 64bit
- 1GB RAM
- BCM43438 wireless LAN and Bluetooth Low Energy (BLE) on board
- 40-pin extended GPIO
- 4 portas USB 2.0
- 4 Saidas de Som Stereo
- 1 entrada Full HD HDMI
- 1 entrada CSI para câmera do Raspberry Pi
- 1 entrada DSI para conectar a Tela Touch para Raspberry Pi
- 1 micro SD para armazenamento e carregar SO
-

A imagem a seguir mostra detalhadamente as especificações encontradas no Raspberry Pi.

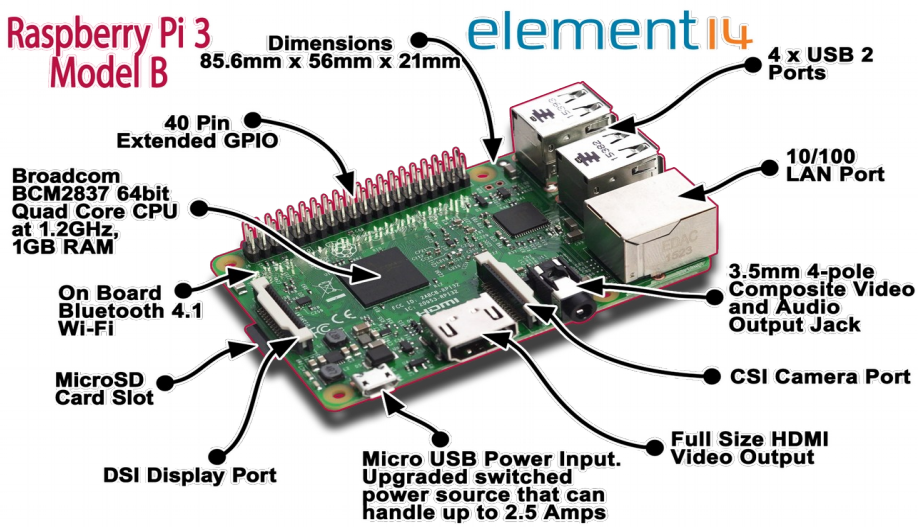

Figura 3: Descrição dos componentes Raspberry Pi 3

[Local: https://www.filipeflop.com/blog/nova-raspberry-pi-3/]

O Raspberry Pi está sendo muito utilizado em projetos que envolvam conectividade entre aparelhos, como ocorre com as cidades inteligentes, casas inteligentes, pesquisar realizadas no campo, como por exemplo, controladores de temperatura, controlador de umidade, sensor de fumaça, entre outros.

Ele possui 40 pinos GPIO possibilitando assim a utilização de vários componentes simultâneos.

Segue mapa da pinagem presente do Raspberry Pi 3 Model B.

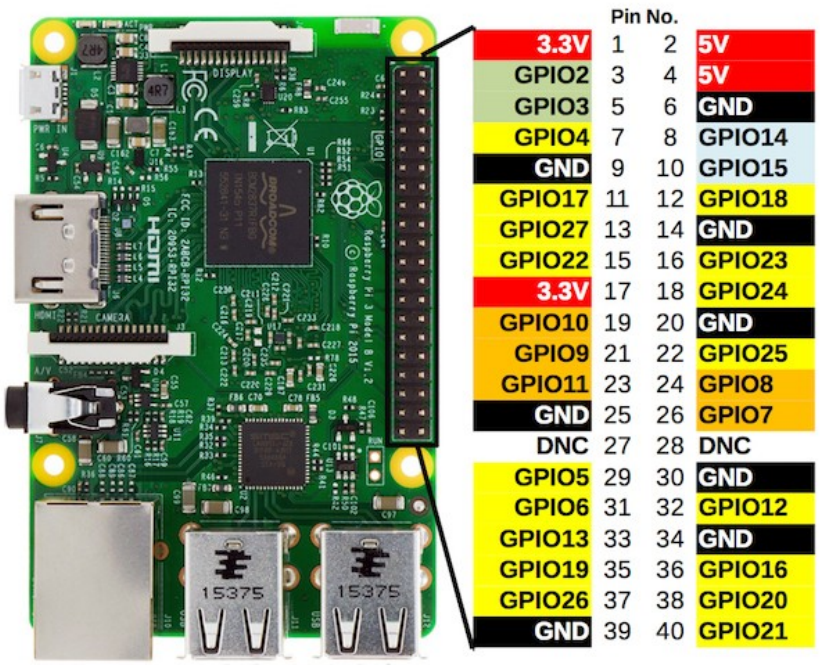

Figura 4: Mapa Pinagem GPIO

[Local: https://www.raspberrypi.org/documentation/usage/gpio/]

No geral, a utilização do Raspberry Pi auxiliado por sensores, nos traz uma nova forma de controlar condições climáticas e ações externas.

3.1. RASPBIAN

O Raspbian é uma distribuição não oficial do Debian que utiliza a arquitetura *Wheezy armhf* otimizado para ser executado no Raspberry Pi.

O próprio Logo da marca é uma junção entre o Logo do Debian e do Raspberry.

Figura 5: Logo Raspbian

[Local: https://www.raspbian.org]

Ele foi produzido por entusiastas da comunidade Raspberry Pi mas levam o nome principalmente de Mike Thompson (mpthompson) e Peter Green (plugwash). (RASPIBIAN.ORG)

Nativamente ele já possuí vários softwares para desenvolvimento e controle sobre o hardware da placa, além de já possuir acesso à Internet e ferramentas de escritório como o LibreOffice.

Tratando-se de um sistema com base em Linux, é de se esperar que seja necessário o aprendizado de alguns comandos básicos no terminal para desfrutar de todas as suas funcionalidades.(RASPIBIAN.ORG)

3.2. TELA LCD (A) RPI 3.5″

O Raspberry Pi possuí vários dispositivos compatíveis para serem utilizados, um deles são as telas LCD, que possibilitam o uso do aparelho em locais variados sem a necessidade de um monitor externo para realizar a interação e exibir as imagens.

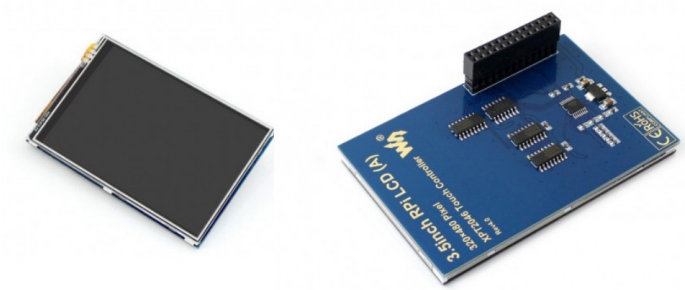

Figura 6: Tela LCD 3.5"

[Fonte: https://www.waveshare.com/wiki/3.5inch_RPi_LCD_(A)]

A tela utilizada nesse projeto tem 3,5” (3,5 polegadas) e *Touch Screen* com uma resolução de 320X400 *Pixels,* dispensando assim a necessidade de um mouse para utilizar as ferramentas fornecidas pelo Raspbian.(WAVESHARE.COM)

A instalação dessa tela em questão é feita através de um pacote disponibilizado pelo fabricante (nesse caso Waveshare) em sua página oficial e depois copie o *driver* para uma pasta em seu SO através dos comandos:

sudo *tar xvf LCD-show-*.tar.gz*
sudo cd LCD-show/

Para transferir a imagem para a tela LCD após ser conectada ao Raspberry Pi deve ser utilizado os comandos *sudo chmod +x LCD35-show* para dar permissão de uso a pasta e logo depois utilizar o comando *./LCD35-show* para utilizar a tela de 3.5″. Caso queira voltar a utilizar a saída HDMI para realizar alguma manutenção é só utilizar o comando ./LCD-hdmi no terminal do Raspbian.(WAVESHARE.COM)

Caso queira saber quais os arquivos de configuração disponíveis de tela LCD é só utilizar o comando *ls* em *cd LCD-show/* como representado pela figura a seguir.

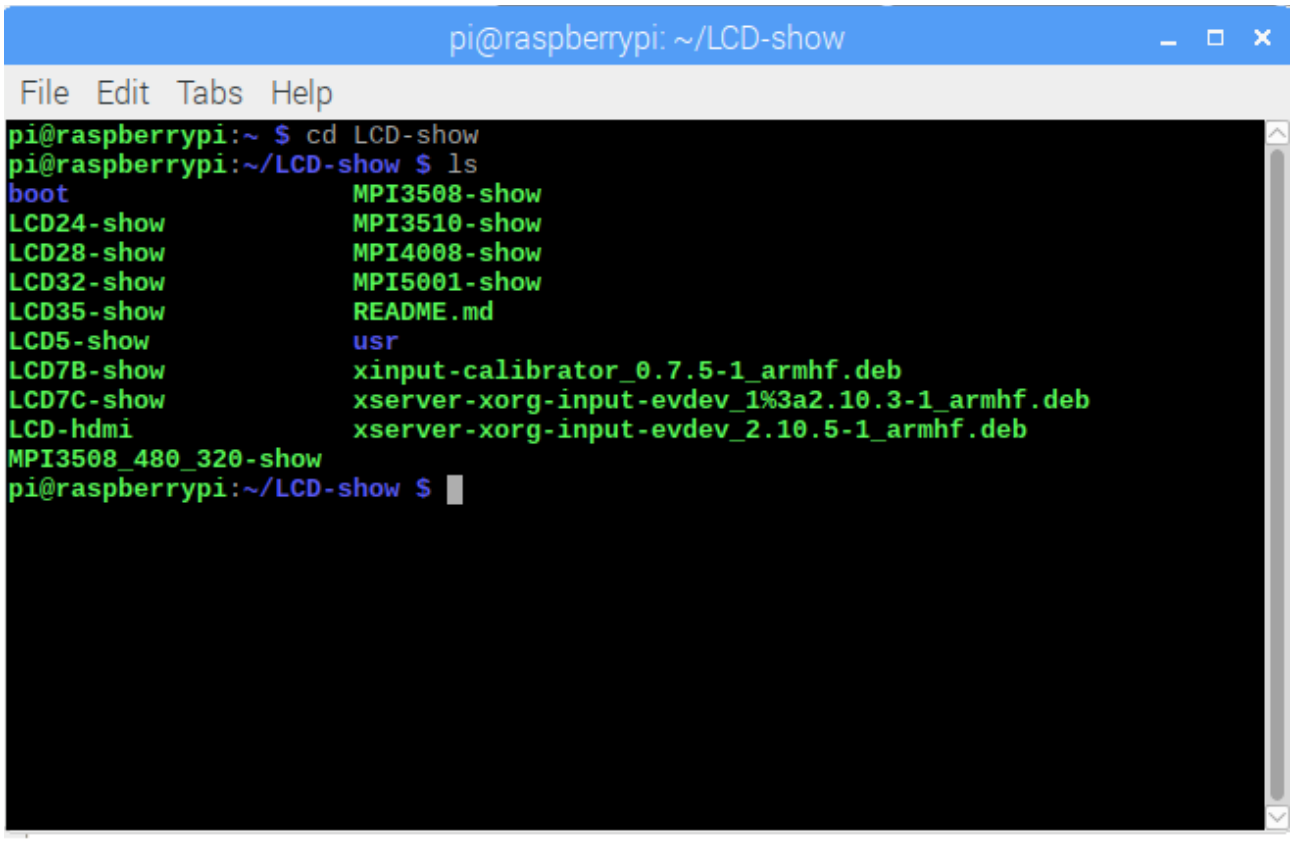

Figura 7: Biblioteca LCD-show

[Fonte: Produção própria]

4. QR CODE

Códigos QR,(*Quick Response*) são códigos de barras Bidimensionais desenvolvido em 1994 pela Denso Wave Corporation, com o intuito de melhorar o controle de tráfego de matérias prima.

Em comparação com códigos de barras normais que só aceitam sua visualização de forma horizontal, eles consegue ser visualizados de várias direções, seja na horizontal ou na vertical e seus códigos são descritos em formato diferenciado.(QRCODE.ORG)

Na imagem a seguir podemos visualizar como é a orientação de ambos os códigos

Figura 8: QR code comparado com um código de barras normal

[Local: http://www.qrcode.com/aboutqr-e.html]

Uma das vantagens do código QR é que ele possui uma grande capacidade de corrigir erros em sua composição, como por exemplo, riscos, sujeira, manchas, etc… Isso porque ele continua conseguindo detectar seu conteúdo mesmo com 30% do código destruído ou apagado.

Seguem exemplos de códigos rasurados que ainda sim podem ser lidos.

Figura 9: Ilustração código sujo e danificado

[Local: http://www.qrcode.com/aboutqr-e.html]

Eles fazem isso através de padrões que são aplicados na estrutura do código, e assim permitindo que eles possam identificar as informações que estão faltando para que a leitura serja feita de forma completa.
Atualmente temos presença desse tipo de código em muitos lugares e sua popularidade cresce a cada dia devido a sua durabilidade e facilidade para realizar sua leitura, pois é possível realizar a leitura através de um smartfone.

Outra vantagem que o código QR possui comparado com o código de barras comum, é a capacidade de informações que podem ser armazenadas.

Os códigos de barras normais consegue guardar um máximo de 20 dígitos enquanto que o código QR pode armazenar até 7.089 caracteres numéricos, 4296 alfanuméricos, 2953 bytes em binário e 1817 caracteres para o Kanji e Kana Japonês.

Outra vantagem do código QR é seu tamanho, que pode ser muito menor do que um código de barras comum e ainda consegue exibir normalmente seus dados armazenados em 1 décimo do espaço normal.

A imagem a seguir demostra o comparativo de tamanhos possíveis entre o código de barras comum e o QR Code.

Figura 10: Comparação tamanho código de barras normal e QR code

[Local: http://www.qrcode.com/en/about/]

Como o código que pode ser visualizado em 360°, ele possui pequenos blocos que servem como detectores de posição, para assegurar uma leitura mais rápida e precisa. (JIMÉNEZ. MARTHA)

Figura 11: Estrutura Código QR

[Local: http://toyoutome.es/pt/blog/el-mundo-en-un-cuadrado-codigos-qr-y-marcadores-ra/20446]

O código é desenhado seguindo alguns padrões que servem para identificar e organizar a forma de como os dados são armazenados.(JIMÉNEZ. MARTHA)

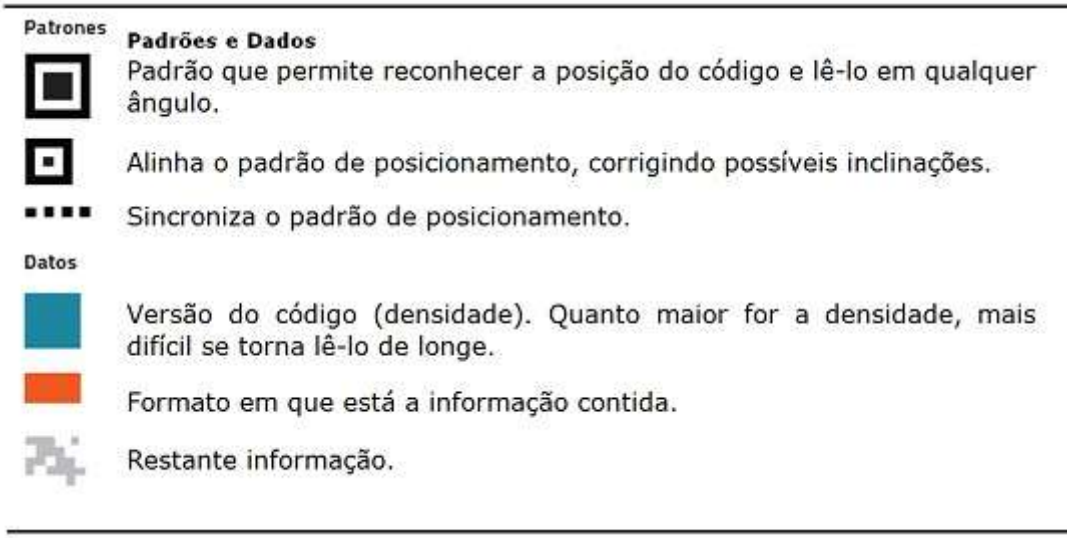

Figura 12: Detalhes padrões Código

[Local : http://toyoutome.es/pt/blog/el-mundo-en-un-cuadrado-codigos-qr-y-marcadores-ra/20446]

Existem muitas formas de analisar ou de construir esses códigos, seja por ferramentas prontas na internet ou também ferramentas construídas a partir de APIs existentes para várias plataformas de programação e linguagens.(QRCODE.ORG)

Um exemplo prático é a biblioteca Z-bar construída em Python que será apresentada mais adiante.

3. LINGUAGEM PYTHON

Python é uma linguagem de alto nível interpretada, orientada a objetos, imperativa, de *script*, funcional e de tipagem dinâmica forte. Ela surgiu em 1991, uma criação do Holandês Guido Van Rossum.

Figura 13: Logotipo Python TM

Local: [www.python.org]

Ele descreveu como foi sua inspiração para criar a linguagem em 1996.

> *"Há mais de seis anos, em dezembro de 1989, eu estava procurando por um projeto de programação como "hobby" que me mantivesse ocupado durante a semana próxima ao Natal. Meu escritório… estaria fechado, mas eu tinha um computador em casa e não muito mais do que isso em mãos. Eu decidi escrever um interpretador para a nova linguagem de scripting sobre a qual eu vinha pensando ultimamente: uma descendente da ABC que agradaria a hackers de Unix/C. Escolhi Python como um título provisório para o projeto, sendo que eu estava num humor um pouco irreverente (e sendo também um grande fã do Monty Python's Flying Circus)."*
>
> ***Guido van Rossum (Maio de 1996)***

Python possui muitos adeptos, dentre eles estão grandes empresas do mundo tecnológico, como, YouTube, o cliente original do BitTorrent, o Google em parte dos *crawlers*(rastreador da rede), Yahoo!, Air Canadá em alguns de seus componentes e até mesmo a NASA utiliza muita linguagem python.

A linguagem prioriza a legibilidade do código sobre a expressividade e velocidade, enfatizando a importância do programador sobre o esforço computacional. Ela combina recursos poderosos de sua biblioteca padrão com módulos e *frameworks* de terceiros..

Por causa de sua capacidade de ser utilizada como linguagem *script*, vários softwares utilizam o Python para adicionar novas funcionalidades e automatizar tarefas, entre eles: BrOffice.org, PostgressSQL, Blender,GIMP e Inkscape.

Python é um software de código aberto compatível com a GPL(*General Public License*) permitindo inclusive que o Python seja incorporado em produtos do tipo proprietário. A especificação da linguagem é mantida pela *Python Software Foundation (PSF)*. O Python em sua compilação traduz o código fonte em *bytecode* e armazena em disco, para que quando ocorrer a próxima execução, não precise compilar novamente o programa e assim, reduzindo o tempo de execução.

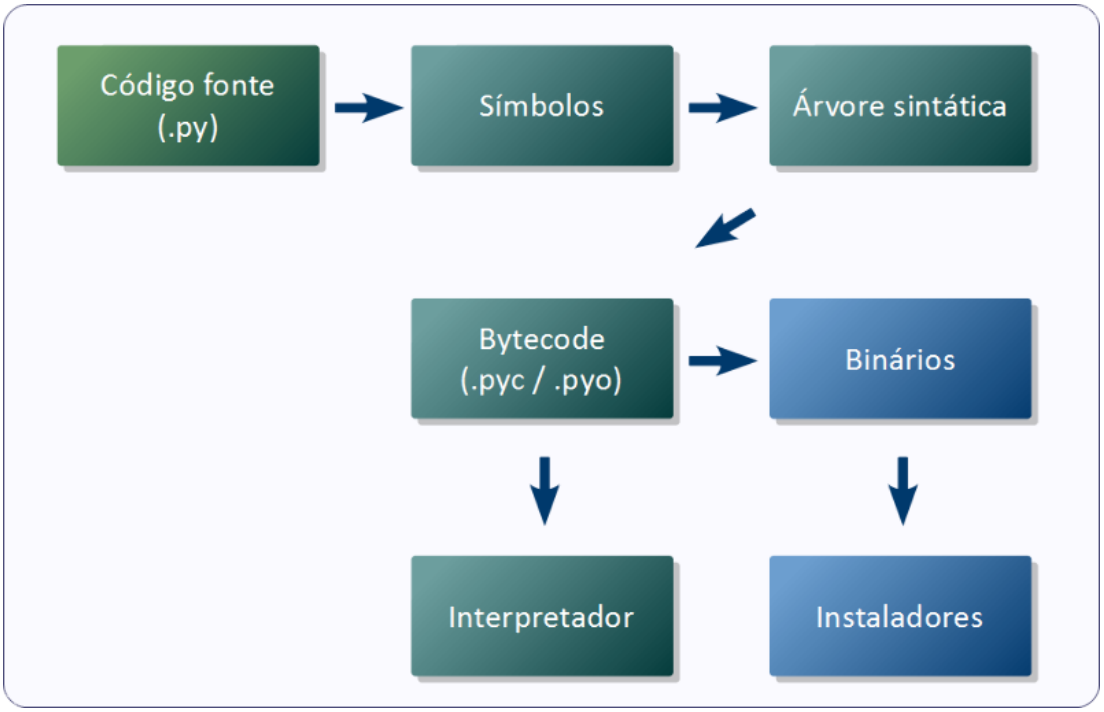

Figura 14: Estrutura compilação Python

[Local: http://ricardoduarte.github.io/python-para-desenvolvedores/]

Caso ocorra alguma alteração, o interpretador que se encarrega de regerar o *bytecode* automaticamente, mesmo se estiver usando um *shell* interativo. O *bytecode* gerado é armazenado com a extensão ".pyc" quando se trata de um *bytecode* normal ou ".pyo" quando é um *bytecode* otimizado.

3.1. THONNY IDE

A Thonny é uma IDE feita para iniciantes na linguagem. Ela está presente no Raspbian como uma das ferramentas nativas para a programação Python.

Ela teve origem no desenvolvimento *open-source* pela comunidade em todo o mundo mas contou com contribuições principalmente do Instituto da Ciência da Computação da Universidade de Tartu, na Estônia e está disponível para Windows, Mac e Linux.

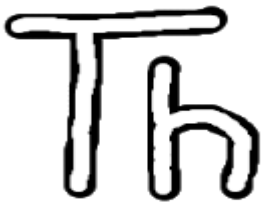

Figura 15: Logotipo Thonny IDE

[Local: https://thonny.org]

Mesmo com um visual bem simples ela possui muitas ferramentas que um usuário iniciante possa querer para o desenvolvimento.

Ele conta com:

- Depurador Simples
- Variáveis sem complicação
- Avaliador Python em suas expressões
- Representação fiel de chamadas de função
- Destaca erros de sintaxe
- Explica Escopos destacando ocorrências de variáveis ajudando a identificar erros de digitação.
- Modo especial para explicar referências.
- Códigos autocompletáveis
- Shell interno para instalação de pacotes extras
- Interface limpa e simples para facilitar o uso de pacotes externos.

3.2. ZBAR-PY

Zbar-py é um módulo que proporciona uma interface de uma biblioteca para leitura de código de barras, com ela é possível ler a maioria dos tipos de códigos de barras, inclusive os códigos QR, possibilitando a leitura de imagens em 2D com vetores do tipo uint8.
Para que ela funcione corretamente é necessárias algumas outras bibliotecas como: “iconv” (necessária para construir a estrutura do Zbar-py), “numpy” para executar o zbar-py e a “pygame” que gerencia o modo de captura das imagens que serão analisadas.

A seguir um exemplo de análise de código de barras em um vetor 2D:

```
import zbar
imagem = ler_imagem_em_array(...) # Qualquer função para
traduzir a imagem em um vetor
scanner = zbar.Scanner()
results = scanner.scan(imagem)
for resultado in results:
 print(resultado.type, resultado.data, resultado.quality,
resultado.position)
```

3.3. TKINTER

Para a construção e manipulação da parte visual do projeto, foi utilizada a biblioteca Tkinter.

A Tkinter é uma biblioteca nativa que permite a construção de interfaces gráficas para o desenvolvedor Python.

Na maioria dos casos, só de possuir o interpretador python instalado na máquina já é possível utilizar essa biblioteca durante a programação.

Caso a ferramenta não encontra-se instalada ela pode ser baixada através do terminal Linux pelo comando sudo apt-get install python-tk ou se estiver utilizando Python3 sudo apt-get install python3-tk.

A escolha da Tkinter para realizar esse projeto foi devido a sua facilidade de uso levando em consideração que ela será desenvolvida para um dispositivo pequeno que não precisa de uma interface muito trabalhada.

Para importar a biblioteca usa-se o comando from Tkinter import *. Com isso todos os componentes da biblioteca já serão carregados.

Sua estrutura básica é formada com a seguinte linha de código:

```
from tkinter import *

root = Tk() #Cria a janela

root.wm_title("Título da janela") #Define o título que aparecerá escrito no todo da janela

root.config(background = "#FFFFFF") #Define a cor de fundo da página, nesse caso branco.



#A programação de todos os Widgets vão nesse espaço



root.mainloop() #Começa a monitorar e atualizar a GUI. Códigos abaixo dessa linha não são executados.
```

A biblioteca possui vários *Widgets* para serem utilizados na construção da interface, os mais utilizados são os:

- Frame(Usado para definir partições dentro da página)
- Label (Para exibir textos escritos na página)
- Entry (Para entrada de dados externos)
- Button (Para adicionar botões de ações)
- Canvas (Utilizado geralmente para representar elementos gráficos como linhas ou textos)
- Text (Para receber ou exibir textos feitos pelo usuário)

4. DECODIFICADOR QR

O projeto consiste em uma *Webcam* ligada a placa do Raspberry Pi onde será feita a captura da imagem do código QR para ser decodificado, uma tela LCD conectada na GPIO da placa, um teclado.

Figura 16: Equipamento utilizado (Produzido pelo autor)

O programa decodificador foi programado em Python utilizando a Thonny IDE presente no próprio Raspbian e sua interface foi desenvolvida utilizando a biblioteca Tkinter gerando a seguinte codificação.

```
#Título da aplicação
w = Label(top, text="Saida de produto!",fg = "light green", bg = "dark green",font = "Helvetica 16 bold italic")
w.pack(side = TOP)
#Botões
B = Button(top, text = "Escanear" , command = start_cam)
B.pack(side = BOTTOM)

B1 = Button(top, text = "Limpar" , command = limpar)
B1.pack(side = 'left')

B2 = Button(top,text = "Gravar" , command = gravar)
B2.pack(side = 'right')

B3 = Button(top,text = "Resumo" , command = resumo)
B3.pack()

L1 = Label(top, text="Quantidade: ", fg = "light green",bg = "dark green",font = "Helvetica 16 bold italic")
L1.pack( side = BOTTOM)
E1 = Entry(top,bd =5)
E1.pack (side = BOTTOM)
E1.focus()


T1 = Text(top,width = 40,height = 1)
T1.configure(state=DISABLED)
T1.pack(side = BOTTOM)

top.mainloop()
```

A imagem a seguir representa a *interface* do programa.

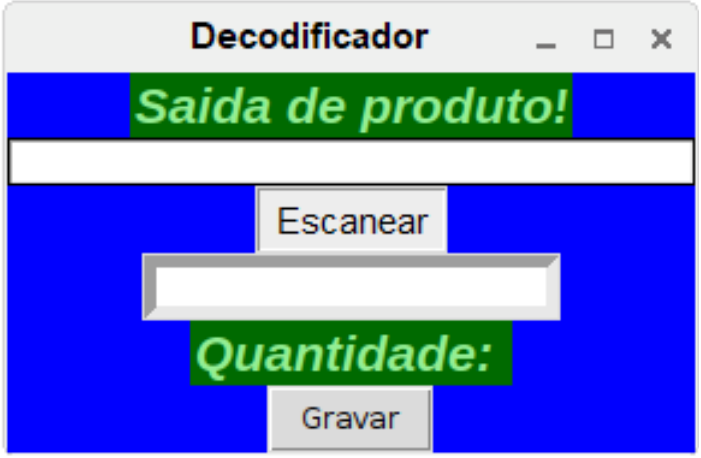

Figura 17: Tela Decodificador QR (Feito pelo autor)

Quando o usuário seleciona o botão “Escanear” é feito a chamada da biblioteca Z-bar que analisa a imagem do QR code capturadas pela câmera e traduz a mensagem criptografada em sua composição.

A seguinte codificação mostra como foi construído esse método

```
def start_cam():

#Inicializa o Zbar na linha de comando para detectar o QRcode.

    p=os.popen('/usr/bin/zbarcam --prescale=300x200','r')

    while True:
#Armazena o codigo lido na barcodedata.

        print("Please Scan a QRcode to begin…")
        barcode = p.readline()
        barcodedata = str(barcode)[8:]
        if barcodedata:

            T1.configure(state=NORMAL)
            T1.insert(INSERT,"{0}".format(barcodedata))
            os.system("/home/pi/kill.sh")

        p.close()
```

A imagem a seguir ilustra a decodificação para a chamada da câmera que realiza a leitura do código.

Figura 18: Leitura do Código QR (Produzido pelo Autor)

A leitura do código é mostrada ao usuário para checar se o item é realmente o que está em mãos e agora é feita a confirmação da quantidade desejada a ser retirada do estoque.

A figura a seguir mostra os detalhes presente na tela do equipamento.

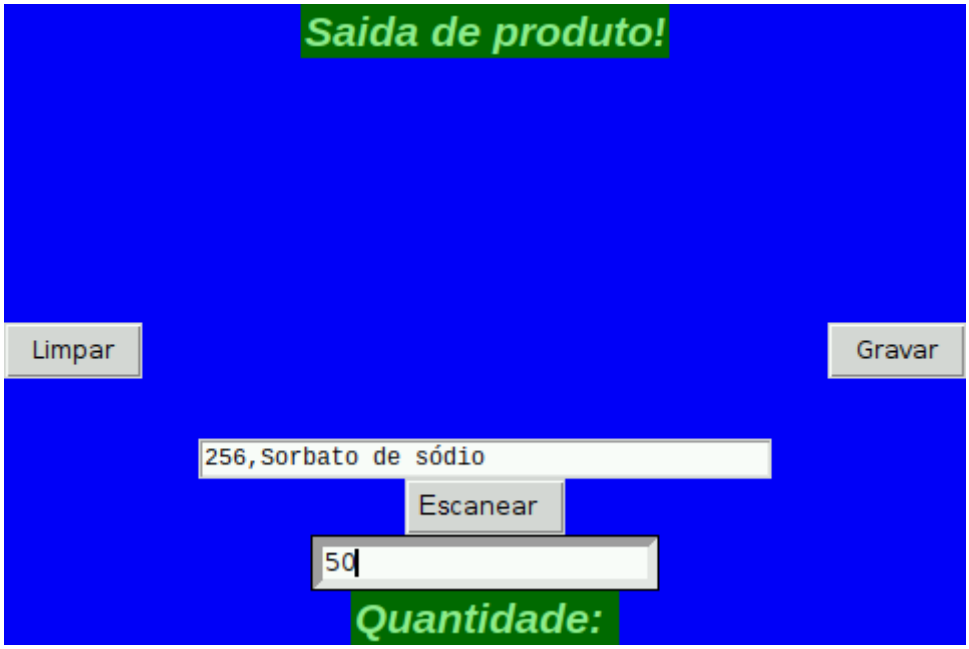

Figura 19: Confirmação de dados (Produzido pelo autor)

Após o usuário pressionar o botão “Gravar” o sistema checa se a quantidade foi digitada ou se o código QR foi escaneado corretamento e mostra uma mensagem de sucesso seguindo a codificação a seguir.

```
def gravar():
f = open("/share/Database.csv","a+")
    conteudo = T1.get("1.0",'end-2c')
    quant = E1.get()

    if conteudo is not "" and quant is not "":
        f.write("{0},{1} \n".format(conteudo,quant))
        messagebox.showinfo("Gravar", "Gravado com sucesso!")
        E1.delete(0,'end')
        T1.delete("1.0",END)
    else:
        messagebox.showinfo("Gravar", "Erro ao gravar, confira os dados!")
    f.close()
    T1.configure(state=DISABLED)
```

A seguir a imagem mostra a confirmação dos dados gravados.

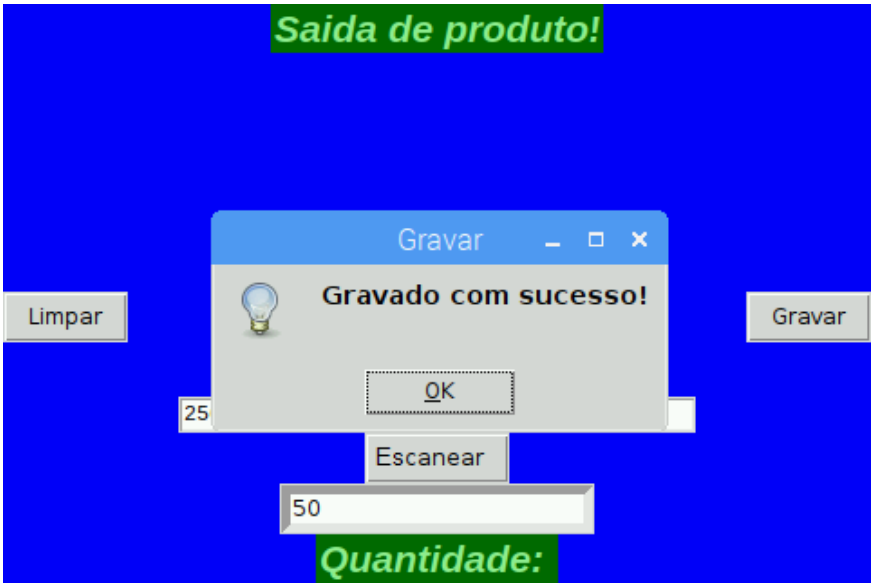

Figura 20: Aviso sucesso na gravação (Produzido pelo autor)

Após a confirmação os dados do QR code e a quantidade do produto informada serão armazenadas em um arquivo CSV e compartilhado em uma pasta na rede para que possa ser acessado do computador administrador que possui o sistema de gerenciamento dos produtos e possa realizar a baixa no estoque.

A seguir um exemplo do arquivo CSV gerado pelo programa.

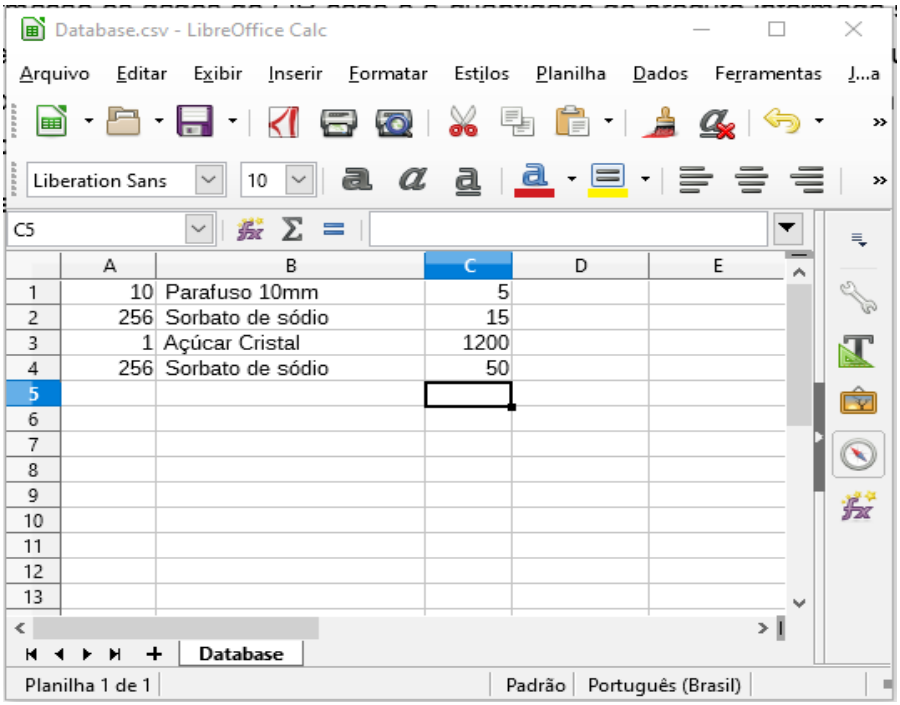

Figura 21: Arquivo CSV que armazena os dados lidos.(Produzido pelo autor)

CONCLUSÃO

O estudo realizado sobre as características do QR Code mostrou que ele tem várias formas de ser utilizado e por ter a capacidade de ser identificado mesmo com rasuras, traz uma maior segurança ao catalogar os itens que vão ser controlados.

O Raspberry Pi se mostrou um equipamento muito abrangente para qualquer setor que eu desejasse implantá-lo, possuindo várias bibliotecas que realizam pequenas coisas mas que em conjunto e com sabedoria podem se tornar fortes ferramentas de aprendizado.

A criação dos códigos QR é bem simples e de forma gratuita o que possibilita testar múltiplas formas de aplicação para checar se todas as possibilidades desejadas são possíveis de ser atendidas.

Quanto a linguagem a ser utilizada, o Python se mostrou muito flexível no quesito de combinações de bibliotecas diferentes.

Ainda será necessário se aprofundar mais para conseguir montar o projeto de forma mais eficiente, para de fato poder ser utilizado com eficácia no fluxo de movimentações mas ele mostrou bem eficiente para realizar o que foi destinado.

REFERÊNCIAS

TECNOBLOG. **5 exemplos de como a automação faz toda a diferença nas empresas.** Disponível em: <https://tecnoblog.net/244245/automacao-solucoes-empresas-exemplos/>. Acesso em: 30 mai. 2018.

ATZORI, Luigi; IERA, Antonio; MORABITO, Giacomo. The Internet of Things: A Survey. Computer Networks,. **Computer Networks**, Itália, v. 1, n. 4247, p. 1-19, mai. 2010.

BMOW.**Raspberry pi gpio programming in c**. Disponível em: <https://www.bigmessowires.com/2018/05/26/raspberry-pi-gpio-programming-in-c/>. Acesso em: 28 mai. 2018.

BORGES, Luiz Eduardo. **Python para desenvolvedores**: Abordando python 3.3. 2 ed. Rio de Janeiro: NOVATEC, 2010. 360 p.

F SHROUF, J ORDIERES, and G MIRAGLIOTTA. **Smart factories in industry 4.0**: A review of the concept and of energy management approached in production based on the internet of things paradigm. In Industrial Engineering and Engineering Management (IEEM), 2014 IEEE International Conference on, pages 697–701. IEEE, 2014.

GONÇALVES, Paulo Sérgio. **Administração de materiais**. 5 ed. Brasil: Campus, 2016. 488 p.

GUBBI, J. et al. A Vision, Architectural Elements, and Future Directions. **Internet of Things (IoT):**, Austrália, v. 1, n. 1, p. 1-19, jul. 2012.

ITF365. **Internet das coisas: desafios que vão além da infraestrutura das redes**. Disponível em: <https://www.itforum365.com.br/gestao/internet-das-coisas-desafios-que-vao-alem-da-infraestrutura-das-redes/ >. Acesso em: 10 mar. 2018.

MÉNDEZ-VILAS, A. et al. **Current developments in technology-assisted education**: Annex. 2 ed. Spain: FORMATEX, 2006. 1504 p.

PYCHARM. **Python ide for professional developers by jetbrains**. Disponível em: <https://www.jetbrains.com/pycharm/ >. Acesso em: 13 mar. 2018.

PYTHON COLOMBIA.**Interfaz gráfica con tkinter**. Disponível em: <https://sites.google.com/site/pythoncolombia/articulos/interfazgraficacontkinter >. Acesso em: 11 mai. 2018.

PYTHON.ORG. **Foreword for "programming python" (1st ed.)**. Disponível em: <https://www.python.org/doc/essays/foreword/ >. Acesso em: 08 mar. 2018.

PYTHON.ORG. **Zbar-py 1.0.4**. Disponível em: <https://github.com/zplab/zbar-py >. Acesso em: 24 jan. 2018.

RASPBERRY.ORG. **Raspberry pi hardware guide**. Disponível em: <https://www.raspberrypi.org/learning/hardware-guide/ >. Acesso em: 16 jan. 2018.

RASPIBIAN. **Raspibian**. Disponível em: <https://www.raspbian.org/raspbianabout >. Acesso em: 18 abr. 2018.

SUSONO, Hitoshi; SHIMOMURA, Tsutomu. Using Mobile Phones and QR Codes for Formative Class Assessment. **Current Developments in Technology-Assisted Education**, Faculty of Education, Mie University, v. 1, n. 1, p. 1-5, jan. 2006.

THE MAGPI MAGAZINE. **Samba: set up a raspberry pi as a file server for your local network**. Disponível em: <https://www.raspberrypi.org/magpi/samba-file-server/ >. Acesso em: 23 mai. 2018.

THONNY PYTHON IDE FOR BEGINNERS. **Thonny python ide for beginners**. Disponível em: <https://thonny.org >. Acesso em: 12 jun. 2018.

TOYOUTOMEBLOG. Proyecto para la transformación digital de la enseñanza. Disponível em: <http://toyoutome.es/pt/blog/el-mundo-en-un-cuadrado-codigos-qr-y-marcadores-ra/20446 >. Acesso em: 04 jul. 2018.

WAVESHARE WIKI. **3.5inch rpi LCD(A)**.Disponível em: <https://www.waveshare.com/wiki/3.5inch_rpi_lcd_(a)>. Acesso em: 16 mai. 2018.