

# ***Sistema de Controle de Consumo Individual por QR Code.***

***Controle de consumo inteligente e organizado a partir de um sistema de leitura de QR Code.***

Luiz Henrique Rocha Marinho  
15/0041527

Engenharia eletrônica.  
Universidade de Brasília  
Brasília – DF, Brasil

[luizhenriquemarinhoFGA@gmail.com](mailto:luizhenriquemarinhoFGA@gmail.com)

Luiza Carneiro Cezário  
15/0041560

Engenharia eletrônica.  
Universidade de Brasília  
Brasília-DF, Brasil

[luiza\\_cezario@hotmail.com](mailto:luiza_cezario@hotmail.com)

**Resumo** —. Este documento possui informações básicas sobre o projeto a ser desenvolvido na disciplina Sistemas Embarcados da Universidade de Brasília, UnB-Gama. O projeto se baseia em um sistema de controle de consumo para estabelecimentos e se baseia na leitura de um QR code gerado individualmente para cada cliente.

**Palavras-chave** — *QR code, Controle de consumo, sistemas embarcados, Raspberry Pi;*

## **I. REFERENCIAL TEÓRICO**

Segundo a ABRASEL (Associação Brasileira de Bares e Restaurantes) o setor alimentício é responsável por cerca de 1 milhão de estabelecimentos no país e gera 6 milhões de empregos diretos [1].

Os estabelecimentos usam diversas formas para controlar o consumo de seus clientes, as formas mais tradicionais são as comandas de papel em posse do cliente ou do garçom, alguns estabelecimentos já utilizam formas mais eficazes de controle de consumo como a comanda com cartão, atendentes com PDA (Personal Digital Assistant) e cardápio através de tablet [2]. A comanda de papel em posse do cliente é vulnerável a falsificações, anotação de pedidos errados e à possibilidade do cliente perder a comanda, a cobrança de taxa

devido a perda da comanda é ilegal pela Lei Federal nº 8.079 do Código de Defesa do Consumidor [3], sendo assim caso perdida ela não pode ser cobrada.

A comanda de papel em posse do garçom além de não ser a opção mais ágil, tem a desvantagem do cliente não ter controle do que está consumindo o que abre a possibilidade para fraudes.

O sistema de atendentes com PDA apresenta uma maior agilidade no processo de atendimento dos clientes, porque o garçom anota os pedidos em seu PDA que são enviados imediatamente para o sistema do restaurante e para a cozinha, melhorando também o controle do restaurante sobre os pedidos. O sistema de comanda por cartão grava no cartão o consumo do cliente no momento em que o pedido é feito o que confere mais rapidez ao processo. O cardápio através de tablet tem função de ser um cardápio interativo que quando o cliente finaliza o pedido ele é diretamente enviado a cozinha, o tablet fica em posse do cliente apenas durante a realização do pedido [2]. A desvantagem dos sistemas por cartão, por atendentes PDA e do cardápio através de tablet é que nesses métodos o cliente também não tem controle sobre seu consumo.

Já existem alguns restaurantes e bares que utilizam a comanda através de Qr Code, em [4] é mostrada uma Startup

que desenvolveu um aplicativo para controle de consumo através de Qr Code.

Qr Code é um código bidimensional de resposta rápida, ele foi criado pelos japoneses e foi inicialmente usado em 1994 na indústria automobilística para controle e catalogação de peças de veículos, atualmente é usado de diversas maneiras como armazenar links de sites, textos, imagens e etc [5]. O Qr Code comparado ao código de barras tem uma capacidade de armazenamento superior, podendo assim carregar mais informações, outra vantagem do Qr Code é que ele é de fácil leitura, muitos smartphones apresentam a função de leitor de Qr Code através da câmera, enquanto que o código de barras precisa de um scanner específico para essa função para ser lido.

## II. JUSTIFICATIVA

O uso das comandas eletrônicas via QR Code visa substituir as comandas tradicionais de papel usadas em diversos estabelecimentos e restaurantes do país.

A comanda tradicional onde o garçom anota os pedidos é facilmente burlada tanto pelo cliente quanto pelo garçom, além de que o processo de atendimento se torna mais lento já que o garçom precisa levar os pedidos até a cozinha.

Com a comanda eletrônica o estabelecimento tem um controle mais eficiente dos pedidos realizados, o processo de atendimento se torna mais ágil porque o pedido feito vai direto para a cozinha e o cliente pode ter controle do seu consumo através da leitura do Qr Code.

## III. OBJETIVOS

Criar um sistema de controle de consumo para restaurantes e bares através da leitura do QR Code individual para cada cliente. O sistema deve ser capaz de gerar Qr Code para cada comanda e ler o código para acessar a interface gráfica onde serão

realizados e registrados os pedidos e calculado o valor da conta, o sistema também deve ser capaz de armazenar os pedidos feitos pelos clientes em um banco de dados para que os dados possam ser recuperados posteriormente em uma nova leitura .

## IV. BENEFÍCIOS

O QR Code é muito utilizado para identificação em dispositivos móveis com câmeras (geralmente celulares)[7], porém esses dispositivos costumam ser caros, utilizando a raspberry PI e uma câmera USB simples, é possível economizar bastante, ponto favorável para estabelecimentos que buscam o lucro.

Controle mais eficiente dos pedidos por parte do estabelecimento evitando fraudes e possíveis erros na realização do pedido.

Em relação ao código de barras, o QR Code possui uma capacidade maior de variações de combinações por conter informações verticais e horizontais [5].

## V. REQUISITOS

- Câmera com conexão USB;
- Gerador de QR Code individual para o cliente no estabelecimento;
- Tela compatível com Raspberry Pi;
- Deve ser de baixo custo
- Precisa apresentar uma interface amigável e de fácil utilização
- Deve ser portátil

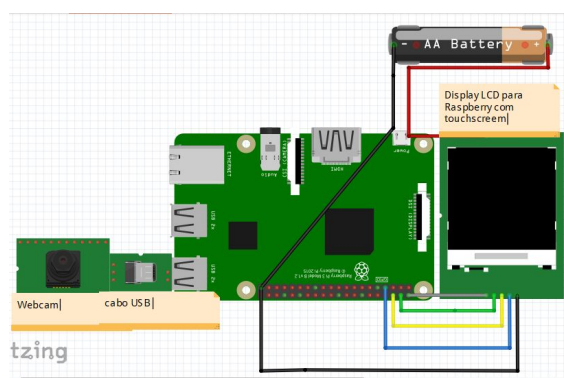
Tabela I : Tabela de Materiais

Unidades	Materiais	Fabricante
1	Raspberry Pi 3 - Model B	Raspberry Pi
1	Web Cam - USB	Clone
1	Display Lcd TFT 3,5 Polegadas	Landio

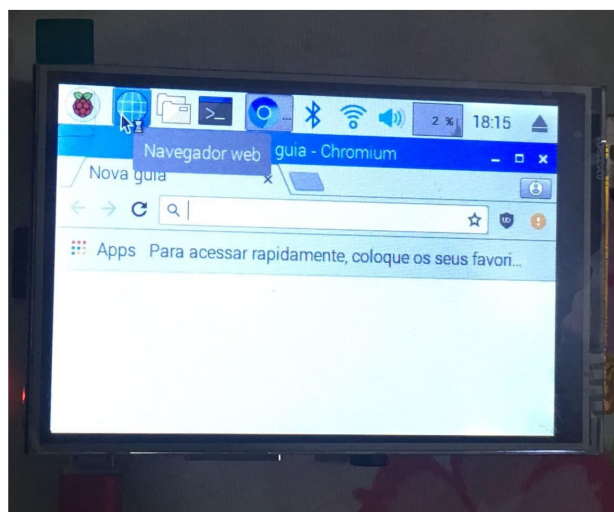
-	Jumpers	-
1	Fonte 5 Volts	-
1	Teclado matricial	-

A partir da biblioteca OpenCV para Python, é possível trabalhar com visão computacional e processamento de imagens [6].

## VI. Hardware



**Figura 1** -Esquemático do projeto realizado no programa Fritzing.

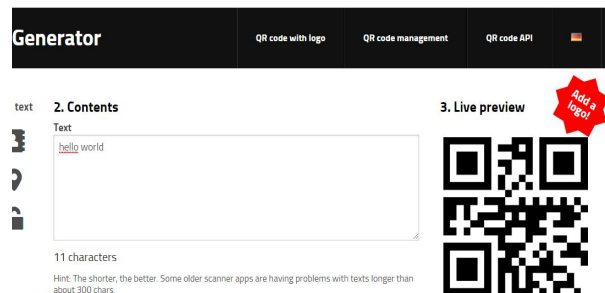


**Figura 2:** Display Lcd

Na figura acima é mostrado o Display lcd em funcionamento que será utilizado no projeto, é um Display LCD TFT Touch de 3,5 polegadas com resolução de 320×480 pixels. O Display ocupa os pinos de 0 a 26 da Raspberry Pi.

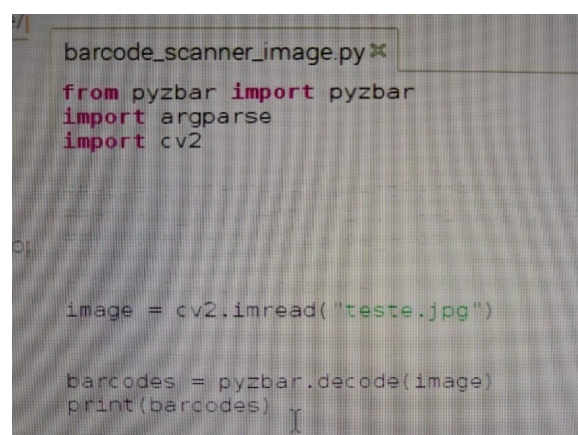
## VII. Software

### A. Leitor de Qr Code



**Figura 2-** QR Code gerado a partir do site <http://goqr.me> o código gerado representa as palavras “hello world”.

A câmera tira foto do Qr Code e a imagem é armazenada e processada com auxílio da biblioteca Open CV que utiliza o módulo pyzbar para decodificar a informação presente no código, após a leitura da informação do Qr Code é possível realizar e gravar os pedidos na comanda digital que armazenará os dados de consumo do cliente, equivalente a comanda manual nos restaurantes.



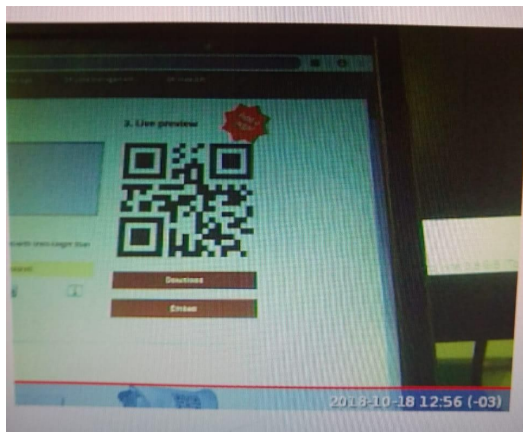
**Figura 3-** Código em Python no terminal para processar o QR Code.

```

Arquivo Editar Abas Ajuda
pi@raspberrypi:~$ source ~/.profile
pi@raspberrypi:~$ workon cv
(cv) pi@raspberrypi:~$ fsw webcam teste.jpg
--- Opening /dev/video0...
Trying source module v4l2...
/dev/video0 opened.
No input was specified, using the first.
Adjusting resolution from 384x288 to 352x288.
--- Capturing frame...
Captured frame in 0.00 seconds.
--- Processing captured image...
Writing JPEG image to 'teste.jpg'.
(cv) pi@raspberrypi:~$ python barcode_scanner_image.py
[Decoded(data='hello world', type='QRCODE', rect=Rect(left=101, top=55,
02, height=104), polygon=[Point(x=101, y=159), Point(x=203, y=156), Point
y=55), Point(x=102, y=57)])]]
(cv) pi@raspberrypi:~$

```

**Figura 4** - Lista de comandos no terminal da Raspberry para ler o Qr Code



**Figura 5** - Imagem obtida a partir do comando “fsw webcam” e com a mensagem “hello world” escrita

### B. Interface gráfica para a comanda

No código da interface gráfica serão cadastrados os produtos que o restaurante oferece para que possam ser inseridos no cardápio digital, a interface vai apresentar uma janela que resume o pedido feito, onde vai ser possível ver: quais foram os produtos consumidos, quantas unidades de cada produto foram consumidas e o valor total do pedido. Caso o cliente queira fazer um novo pedido o Qr Code da comanda será escaneado novamente e poderão ser adicionados ao pedido novos produtos mantendo também os produtos consumidos do pedido anterior.

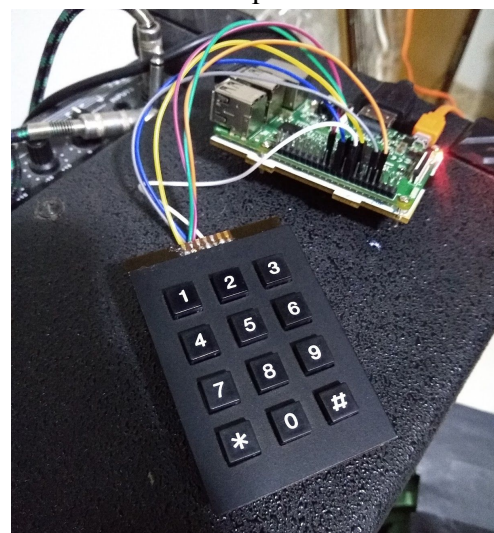
```

Restaurate
*****
* Cardápio *
* Item      Preço *
* 1.Batata Frita      R$ 8,00 *
* 2.Coca Cola         R$ 5,00 *
* 3.Pizza de Calabresa R$ 30,00 *
* 4.Cachorro Quente   R$ 5,00 *
* 5.Suco              R$ 5,50 *
* 6.Crepe de Frango   R$ 7,00 *
* 7.X-Burguer         R$ 7,00 *
* 8.Tapioca           R$ 5,00 *
* 9.Marmita           R$ 10,00 *
* 10.Açaí             R$ 5,00 *
*****
Digite os números dos itens que deseja e a
quantidade da seguinte forma: 01#02# Para ter-
minar digite: *

```

**Figura 6:** Exemplo da interface com cardápio genérico

A prévia da interface gráfica representada na imagem acima foi feita em linguagem C, até o momento ela ainda não apresenta design, conta apenas com uma apresentação básica no terminal, foram utilizadas as bibliotecas stdio.h e stdlib.h para o desenvolvimento do código da comanda. Até o momento programa é capaz de receber os pedidos do usuário e receber a quantidade de itens, mostrar o preço dos produtos e o preço total do pedido, um teclado matricial será usado para indicar o item e a quantidade.



**Figura 7:** teclado matricial 3x4.

O programa é capaz também de salvar em arquivo os itens do pedido, que é onde será salvo os dados das comandas e

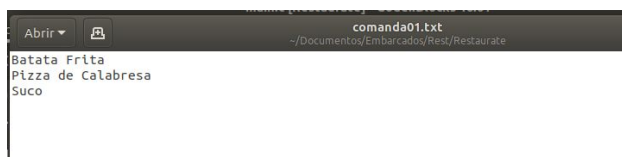


será desse arquivo que serão retomados os dados caso o cliente queira acrescentar algo no pedido ou olhar o que já foi consumido.



**Figura 8:** Exemplo de pedido

Para realizar o pedido o usuário digita o número do item, logo em seguida # e a quantidade de produtos do mesmo item que ele deseja, para finalizar o pedido ele deve digitar \*. Logo em seguida aparece o nome dos itens pedidos o total em reais correspondente a cada item e o preço total do pedido. Ao ser finalizado os itens serão salvos em um arquivo destinado a comanda.



**Figura 9:** Produtos salvos na comanda

Até o momento apenas os itens estão sendo salvos na comanda, ainda é preciso definir a função para salvar as quantidades e o preço total.

```

char pedido[500];
FILE *fp;

fp = fopen("comanda01.txt", "w");
if(!fp)
{
    printf( "Erro na abertura do arquivo");
    exit(0); }
  
```

**Figura 10:** Função para criar o arquivo da comanda

Nos próximos pontos de controle o nome da comanda será fornecido pelo Qr code, neste ponto de controle a comanda destinada foi previamente definida.

```

else if(itemnum[b]==3)
{
    char string[20] = "Pizza de Calabresa\n";
    printf(string);
    float preco = quantit[b]*30;
    printf("%f\n", preco);
    for(i=0; string[i]; i++) putc(string[i], fp);
    precotot = precotot + preco;
}
  
```

**Figura 11:** Função para escrever na comanda

A função acima escreve no terminal o item escolhido pelo usuário, o preço do pedido e escreve no arquivo da comanda o nome do item pedido.

O código do teclado matricial foi desenvolvido em C com o auxílio da biblioteca wiringPI.h que facilita a utilização dos pinos da raspberry pois trabalha com uma numeração mais intuitiva. o código encontra-se anexado no fim do documento.

### C. Funcionamento do sistema:

Quando o cliente pedir algum item do cardápio, o garçom deve escanear o Qr code da comanda através da câmera, quando o Qr code for processado a interface será apresentada na tela lcd onde o garçom poderá selecionar os produtos escolhidos pelo cliente e realizar o pedido, será apresentado também a lista de produtos consumidos e o valor total do

pedido, ao terminar o pedido os dados serão armazenados em uma base de dados interna para que quando o garçom escanear o código novamente os dados do pedido possam ser recuperados.

## VIII. Resultados

Até o presente relatório a dupla conseguiu desenvolver de maneira satisfatória o código do leitor de Qr Code, sendo capaz de armazenar o Qr Code e decodificá-lo, a dupla também foi conseguiu desenvolver parte da lógica da interface em linguagem C e conseguiu escrever os pedidos na comanda, também foi desenvolvido o código do teclado em python e em C. O teclado em python está funcionando satisfatoriamente, em C ainda faltam alguns ajustes para que o teclado funcione corretamente, mas boa parte da lógica já foi desenvolvida. A dupla também conseguiu configurar corretamente e deixar pronto para uso o display LCD que será usado para apresentar a interface.

Para o próximo ponto de controle espera-se desenvolver uma interface gráfica com melhor design e usabilidade com o auxílio de bibliotecas em linguagem C ou C++, e fazer integração dos códigos da comanda e do código que lê o Qr Code e espera-se finalizar também o código do teclado em linguagem C.

## IX. REFERÊNCIAS

- [1] Abrasel. “Perfil da Abrasel”. 2018. <http://www.abrasel.com.br/perfil-da-abrasel.html>
- [2] Ferronato, A. “Pila Fácil: Sistema de Gerenciamento de Pedidos”. Porto Alegre, Universidade Federal do Rio Grande do Sul, 2010.
- [3] CONSUMIDOR. Código do Consumidor. LEI Nº 8.078, de 11 de Setembro de 1990. [http://www.planalto.gov.br/ccivil\\_03/Leis/L8078.htm](http://www.planalto.gov.br/ccivil_03/Leis/L8078.htm)
- [4] G1.com. “Startup cria comanda digital para evitar filas em bares e restaurantes”. 2018. <https://g1.globo.com/economia/pme/pequenas-empresas-grandes-negocios/noticia/2018/09/02/startup-cria-comanda-digital-para-evitar-filas-em-bares-e-restaurantes.ghtml>
- [5] QRcode.com. “About QR Code”. 2012. <http://www.qrcode.com/en/aboutqr.html>.
- [6] Bradsky, G. R.; Pisarevsky, V.; Bouguet, J. Learning OpenCV: Computer Vision with the OpenCV Library. Springer, 2006.
- [7] Silva, P. C. Neto, Nunes, C. e Nunes, E. P. dos Santos. “Integrando Recursos de Realidade Aumentada e Código de Barras Bidimensionais no Desenvolvimento de um Guia de Turismo”. In WRVA 2011. 2011.

Anexos:

```
Arquivo Editar Procurar Opções Ajuda
#include <stdio.h> //inclui o cabeçalho da biblioteca padrão de C.
#include <wiringPi.h> //inclui a biblioteca wiringPi.

int led = 0;

int linha_1 = 1;
int linha_2 = 2;
int linha_3 = 3;
int linha_4 = 4;

int saida_1 = 5;
int saida_2 = 6;
int saida_3 = 7;

int main(void) { //inicia a função principal
wiringPiSetup(); //inicializa o wiringPi
pinMode(0, OUTPUT); //define o modo do pino como saída.

pinMode(1, OUTPUT);
pinMode(2, OUTPUT);
pinMode(3, OUTPUT);
pinMode(4, OUTPUT);

pinMode(5, INPUT);
digitalWrite(5, HIGH);

pinMode(6, INPUT);
digitalWrite(6, HIGH);

pinMode(7, INPUT);
digitalWrite(7, HIGH);

while (1) { //loop
for (int porta = 1; porta<5; porta++)
{
//Alterna o estado dos pinos das linhas

digitalWrite(1, HIGH);
digitalWrite(2, HIGH);
digitalWrite(3, HIGH);
digitalWrite(4, HIGH);
digitalWrite(porta, LOW);

//Verifica se alguma tecla da coluna 1 foi pressionada
if (digitalRead(5) == LOW)
{
digitalWrite(led, HIGH); //acendo o LED.
delay(1000);
digitalWrite(led, LOW); //apaga o LED.
}
```

**Anexo 1:** código em C do teclado matricial parte 1.

```
Arquivo Editar Procurar Opções Ajuda
{
digitalWrite(led, HIGH); //acendo o LED.
delay(1000);
digitalWrite(led, LOW); //apaga o LED.
while(digitalRead(5) == LOW){};

}

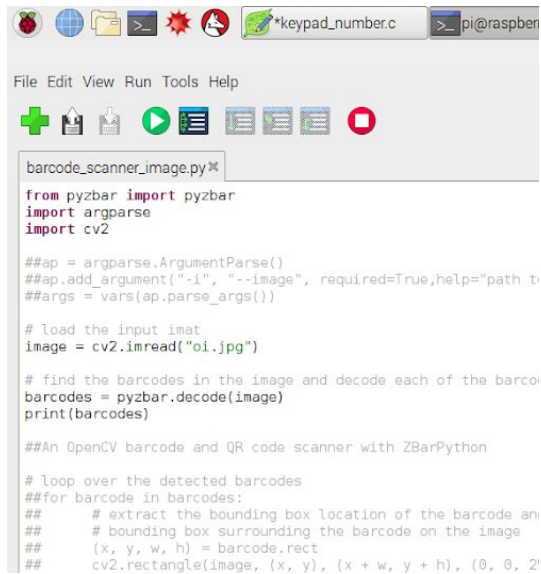
//Verifica se alguma tecla da coluna 2 foi pressionada
if (digitalRead(6) == LOW)
{
digitalWrite(led, HIGH); //acendo o LED.
delay(1000);
digitalWrite(led, LOW); //apaga o LED.
delay(1000);
digitalWrite(led, HIGH); //acendo o LED.
delay(1000);
digitalWrite(led, LOW); //apaga o LED.
while(digitalRead(6) == LOW){};

}

//Verifica se alguma tecla da coluna 3 foi pressionada
if (digitalRead(7) == LOW)
{
digitalWrite(led, HIGH); //acendo o LED.
delay(1000);
digitalWrite(led, LOW); //apaga o LED.
delay(1000);
digitalWrite(led, HIGH); //acendo o LED.
delay(1000);
digitalWrite(led, LOW); //apaga o LED.
while(digitalRead(7) == LOW){};

}
}
delay(10);
}
return 0;
}
```

**Anexo 2:** código em C do teclado matricial parte 2



The image shows a terminal window on a Raspberry Pi. The title bar includes icons for a Raspberry Pi, a globe, a folder, a terminal, a red star, a red circle with a white 'X', and a file named 'keypad\_number.c'. The terminal window has a menu bar with 'File', 'Edit', 'View', 'Run', 'Tools', and 'Help'. Below the menu bar is a toolbar with icons for a green plus, a document, a magnifying glass, a play button, a list, a document with a magnifying glass, a document with a magnifying glass and a red circle, and a red circle with a white 'X'. The terminal content shows a Python script named 'barcode\_scanner\_image.py' with the following code:

```
barcode_scanner_image.py
from pyzbar import pyzbar
import argparse
import cv2

##ap = argparse.ArgumentParser()
##ap.add_argument("-i", "--image", required=True, help="path to image")
##args = vars(ap.parse_args())

# load the input image
image = cv2.imread("oi.jpg")

# find the barcodes in the image and decode each of the barcodes
barcodes = pyzbar.decode(image)
print(barcodes)

##An OpenCV barcode and QR code scanner with ZBarPython

# loop over the detected barcodes
##for barcode in barcodes:
##    # extract the bounding box location of the barcode and
##    # bounding box surrounding the barcode on the image
##    (x, y, w, h) = barcode.rect
##    cv2.rectangle(image, (x, y), (x + w, y + h), (0, 0, 255), 2)
```

**Anexo 3:** comando em python para realizar o processamento do QR code e retornar apenas a informação que interessa para o projeto: a informação do QR code.