Universidade Federal do Estado do Rio de Janeiro

PROGRAMA DE PÓS-GRADUAÇÃO MATEMÁTICA EM REDE NACIONAL

MESTRADO PROFISSIONAL EM MATEMÁTICA EM REDE NACIONAL

A MATEMÁTICA DO CÓDIGO DE BARRAS E QR CODE

ALBERTO RENAN DIAS DA SILVA

Rio de Janeiro

Agosto, 2021

Título: A MATEMÁTICA DO CÓDIGO DE BARRAS E QR CODE.

Dissertação apresentada ao Programa de Pós-graduação em Matemática PROFMAT da UNIRIO, como requisito para a obtenção do grau de MESTRE em Matemática.

Orientador: Silas Fantin
Professor Doutor em Matemática
Unirio

Rio de Janeiro
Agosto, 2021

Alberto Renan Dias da Silva

A MATEMÁTICA DO CÓDIGO DE BARRAS E QR CODE

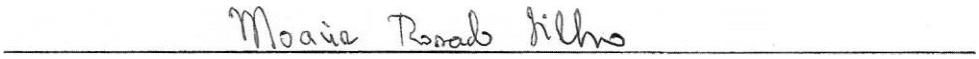
Dissertação apresentado ao Programa de Pós-graduação em Matemática PROFMAT da UNIRIO, como requisito para a obtenção do grau de MESTRE em Matemática.

Aprovada em 12 de agosto de 2021.

BANCA EXAMINADORA

Prof. Dr. Silas Fantin — Orientador
UNIRIO

Prof. Dr. Michel Cambrainha de Paula
UNIRIO



Prof. Dr. Moacir Rosado Filho
UFES

Rio de Janeiro
2021

RESUMO

O objetivo deste trabalho é mostrar que os códigos podem ser utilizados nas aulas de matemática básica com a finalidade de fazer os alunos se interessarem pelo tema. Vamos mostrar as características do código de barras e do qr code e como eles podem se relacionar com os conteúdos matemáticos. Por fim, mostraremos atividades para aplicação em sala de aula que envolvam o tema deste trabalho.

Palavras-chave: códigos de barras, qr codes, matemática, atividades.

ABSTRACT

The objective of this work is to show that the codes can be used in basic math classes as a way to make the students become interested in the subject. We will show the characteristics of the barcode and the QR Code and how it can relate to mathematical content. Finally, we will show activities for application in the classroom that involve the theme of this work.

Keywords: bar codes, qr codes, mathematics, activities.

AGRADECIMENTOS

Primeiramente gostaria de agradecer a Deus por todas as realizações em minha vida.

Neste meu caminho de estudos sempre tive total ajuda da minha família e de minha esposa que me incentivaram a nunca desistir dos meus sonhos então gostaria de agradecê-los por todo o esforço e companheirismo.

Agradeço ao meu orientador Prof. Dr. Silas por toda ajuda e paciência ao longo deste trabalho.

Catalogação informatizada pelo(a) autor(a)

S586 SILVA, ALBERTO RENAN DIAS DA
A MATEMÁTICA DO CÓDIGO DE BARRAS E QR CODE / ALBERTO RENAN DIAS DA SILVA. -- Rio de Janeiro, 2021.
69

Orientadora: Silas Fantin.
Dissertação (Mestrado) - Universidade Federal do Estado do Rio de Janeiro, Programa de Pós-Graduação em Matemática, 2021.

1. QR Code. 2. Código de Barras. 3. Números binários. I. Fantin, Silas, orient. II. Título.

SUMÁRIO

LISTA DE FIGURAS

LISTA DE TABELAS

INTRODUÇÃO

Este trabalho tem como objetivo apresentar uma proposta de abordagem sobre o Código de Barras e o QR Code para professores da Educação Básica, com o propósito de inseri-lo nas aulas de matemática trazendo seu contexto histórico e cultural como forma de um agente lúdico para melhor desenvolvimento dos conteúdos e vamos mostrar algumas aplicações deles no dia a dia. “Utilizar dispositivos móveis (celular) e o aplicativo QR Code como recurso pedagógico para potencializar o ensino e aprendizagem da matemática” (Pinto, 2016, p. 5).

A ideia é criar uma linha de conhecimento acerca do Código de Barras e QR Code para embasar o foco deste trabalho que será um conjunto de atividades que poderão ser desenvolvidas em sala de aula do Ensino Básico, que, segundo Silva 2016, pode diminuir a dificuldade dos alunos com relação a matérias ministradas pelos educadores.

No primeiro capítulo, apresentaremos um apanhado da história dos objetos de estudo deste trabalho, onde foram as primeiras aparições, seus tipos, versões e nomes.

No segundo capítulo, iremos trabalhar a matemática básica por trás dos códigos de barras, como o cálculo do digito verificador e a conversão de barras em números.

No terceiro capítulo, a ideia é mostrar todos os elementos embutidos em um QR Code, dando foco na versão 1 (21 x 21) que será apresentada nos próximos capítulos.

A seguir, no quarto capítulo iremos mostrar um conjunto de atividades dos grandes vestibulares nacionais que envolvam os códigos trabalhados aqui e suas resoluções e, por fim, algumas atividades desenvolvidas pelo autor para serem aplicadas em sala de aula.

Para Sadovsky (2007):

> “[...] a Matemática, não só no Brasil, é apresentada sem vínculos com os problemas que fazem sentido na vida das crianças e dos adolescentes. Os aspectos mais interessantes da disciplina, como resolver problemas, discutir ideias, checar informações e ser desafiado, são pouco explorados na escola. O ensino se resume a regras mecânicas que ninguém sabe, nem o professor, para que servem. ”

Com base no supracitado e tendo em vista a crescente aparição do QR Code na sociedade moderna e que o código de barras faz parte de nossas vidas há muito tempo e se torna necessário que essas tecnologias cheguem à sala de aula e agreguem conteúdo às aulas, principalmente de matemática e informática, tanto para o desenvolvimento dos assuntos como para a formação dos discentes e, com isso, a sua inserção profissional na sociedade de maneira satisfatória. Para Pinto (2016):

“Um dos objetivos em sala de aula é contribuir para a formação de cidadãos pensantes, críticos, reflexivos e motivados a discutir problemas e aprofundar os conhecimentos”.

Enfim, o objetivo do trabalho não era criar um acumulado de informações dispersas sobre os códigos estudados, mas criar uma base de conhecimento sobre eles para que professores possam ter um fundamento ao aplicar as atividades em sala de aula e mostrar como eles podem ser inseridos em algumas aulas de forma a acrescentar nos sentidos histórico e aplicado da matemática e criando alternativas de trabalhos e atividades, mostrando um outro olhar sobre a matemática.

1. A HISTÓRIA DO CÓDIGO DE BARRAS E DO QR CODE

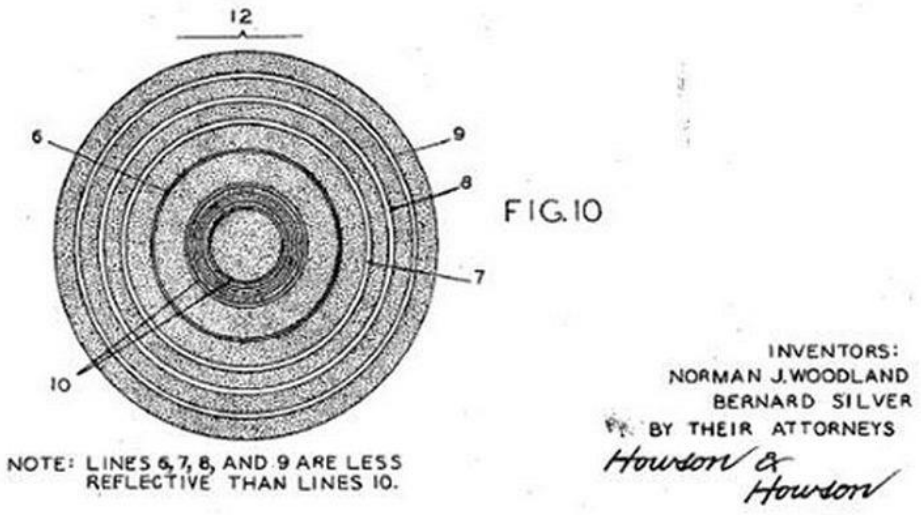
A história do Código de Barras começa há mais de 70 anos onde a necessidade de automação na hora da venda de produtos em supermercados fez com que o CEO de uma grande rede procurasse um aluno do Instituto de Tecnologia Drexel (atual Universidade Drexel), na Filadélfia em 1948. A ideia era ter uma identificação do produto na hora em que o mesmo passasse no caixa facilitando e agilizando as vendas.

Segundo o site GS1BR, o aluno Bernard Silver ouviu nos corredores do Instituto o CEO pedir a outro aluno um método de automação para capturar as informações de determinado produto quando ele é passado no caixa.

Silver pensou sobre a proposta e discutiu acerca com Joseph Woodland, também aluno do Instituto, que gostou da ideia e decidiu desenvolvê-la. No começo, pensaram em um método que utilizava uma tinta reflexiva que brilhava quando exposta a luz ultravioleta. A ideia foi um sucesso, mas os alunos tiveram dois problemas: alto custo da tinta e instabilidade da mesma.

Após um bom tempo desenvolvendo seu método, Woodland conseguiu chegar ao que hoje chamamos de Código de Barras através da união do Código Morse e da Banda Sonora. Mas o primeiro código era bem diferente do utilizado atualmente, era formado por círculos concêntricos.

Figura 1: Primeiro código.



Fonte: i9automacaocomercial [1], 2021.

Woodland e Silver apresentaram a patente do código e, em 1952, foi-lhes concedido registro que foi posteriormente vendido para a empresa Philco por US$ 15,000, em seguida para a empresa RCA e expirou-se em 1969.

Os colegas continuaram trabalhando em sua ideia, mas agora o desafio era outro, pois a tecnologia da época não permitia a construção de um pequeno aparelho que fizesse a leitura do código e que, acima de tudo, fosse barato.

Apenas em 1970, com a tecnologia avançando é que foi possível desenvolver um aparelho com as especificações desejadas com a ajuda do laser. Woodland trabalhava na IBM no Estado da Carolina do Norte, onde pôde testar o laser no código, que agora era de listras pretas e brancas, e a leitura era feita na reflexão da luz nas listas que era captada pelo aparelho que conseguia interpretar a informação contida no código.

E foi em 1974 que o primeiro produto foi comprado por um cliente em um caixa de supermercado onde o Código de Barras era registrado, era um pacote de pastilhas, que foi passado pelo scanner pela operadora de caixa e houve a leitura do código, indicando as características e o valor do produto.

Segundo o site GS1, hoje em dia existem diversos tipos de códigos sendo o EAN o mais conhecido, mas o emprego de cada um depende do tipo e quantidade de informação a ser armazenada dentro do código. Alguns deles são:

EAN/UPC

É o tipo mais comum na identificação de bens de consumo em geral. O código EAN possui 13 dígitos e um tamanho padrão, enquanto o UPC possui 12 dígitos.

O EAN é o modelo utilizado no Brasil.

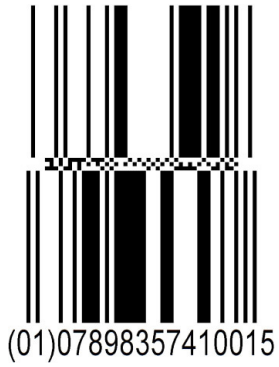
Figura 2: EAN/UPC.

Fonte: gs1br [2], 2021.

GS1 DataBar

É um código composto por duas fileiras de listras, uma acima da outra, e mesmo sendo menor que o EAN/UPC consegue armazenar mais informações.

Figura 3: GS1 DataBar.



Fonte: gs1br [2], 2021.

Código 128

Este código tem a sua utilização bem específica em indústrias e transportes por ser bem compacto e denso, permitindo a armazenagem de muitas informações.

Figura 4: Código 128.

Fonte: gs1br [2], 2021.

Intercalado 2 de 5

Também conhecido como ITF-14, é bem comum na área de transportes e por operadores logísticos. Ele é bem compacto e pode ser encontrado em embalagens e caixas de papelão, usadas para acondicionar os produtos, que serão enviados para os intermediários (como distribuidores).

Figura 5: Intercalado 2 de 5.

Fonte: Bis WikiDocs [3], 2021.

QR CODE

É um código bidimensional muito utilizado hoje em dia e possui múltiplas aplicações no mercado desde gerência logística, armazenagem de links de sites, e qualquer outra informação alfanumérica.

Ele foi criado em 1994 pela empresa Denso Wave, uma subsidiária da Toyota, que produzia peças de automóveis, com o intuito de ter um código à mão que pudesse conter as características das peças produzidas.

Uma grande diferença entre o código de barras e o QR Code é que o segundo código possui a licença livre na qual qualquer empresa pode utilizá-lo livremente ao contrário do código de barras que, como já foi citado, é gerenciado por uma empresa no mundo todo.

Segundo o site gs1br (2021):

> "O foco deste padrão é na Embalagem Estendida do produto, ou seja, por meio de uma URL informada, mostra informações que não estão visíveis no produto já identificado com padrão GS1, propiciando aos consumidores acesso às informações adicionais ou aos serviços referentes aos produtos, inclusive permitindo redirecionamento para conteúdo do website do produto ou empresa. "

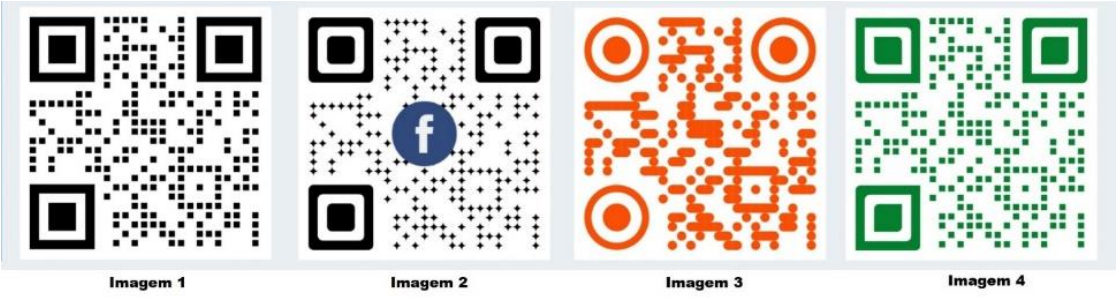
Os QR Codes possuem dois modelos denominados "model 1" e "model 2", sendo que o segundo é dividido em vários tamanhos que são chamados de versões. Essas versões podem ir de 21 x 21 pixels (versão 1) até 177 x 177 pixels (versão 40) e essas numerações indicas a quantidade de pixels que o código irá conter como, por exemplo, no caso da versão 1 teremos 21 x 21 = 441 quadradinhos que podem variar apenas entre duas cores, preta ou branca.

Assim como no código de barras, o QR Code também precisa de um software leitor para decodificar a mensagem contida na imagem, hoje em dia esse software está contido nas câmeras da maioria dos celulares que são fabricados, mas os usuários de smartphones também podem baixá-los nas lojas de aplicativos.

Alguns exemplos de aplicativos para smartphones, grátis e disponíveis para leitura e escrita dos códigos, são "Leitor de código de barras e QR", "Leitor de código QR e Leitor de código de barras" e "Leitor de código QR e Leitor de código de barras" e, além disso, as câmeras dos celulares modernos já possuem um leitor de QR Code de fábrica.

Este novo código se tornou algo tão versátil que permite a introdução de imagens como logomarcas no interior sem a perda, parcial ou total, da capacidade de leitura, outro detalhe é que mesmo modificando as cores ou formatos dos componentes do QR Code padrão, o software leitor conseguirá realizar a leitura perfeitamente.

Figura 6: Novos QR Codes.



Fonte: autor.

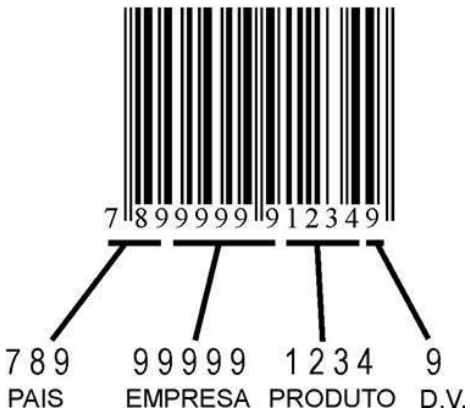
Na figura acima temos quatro QR Codes contendo a mesma mensagem, o número 1234567890 que é formado por todos os algarismos. O detalhe é que na primeira imagem temos um QR Code padrão, ou seja, sem modificações, já nas outras três imagens temos QR Codes modificados, com cores diferentes e até a aparição de uma logomarca. Esta nova forma de codificação vem ganhando espaço na sociedade na última década e sendo utilizada por lojas e empresas nas suas campanhas publicitárias pela facilidade de armazenar URLs, senhas de wi-fi, textos, números de telefone, e-mails, redes sociais, localizações em mapas e muito mais. Na TV é comum que apareça em programas, promoções e campanhas publicitárias para serem escaneados e o usuário redirecionado para algum site.

2. COMO FUNCIONA O CÓDIGO DE BARRAS

Neste capítulo vamos mostrar a lógica por trás do Código de Barras utilizado mundialmente nos comércios, para isso escolhemos o código utilizado no Brasil, o EAN (European Article Numbering System) ou GTIN-13.

Como foi mostrado no capítulo 2, o EAN, que no Brasil é chamado de GTIN-13, é utilizado tanto nas lojas físicas, quanto no e-commerce para vendas online, possui 13 algarismos que são separados em três partes da seguinte forma X-XXXXXX-XXXXXX. Temos o primeiro dígito que fica na parte esquerda e externa da figura e mais duas partes contendo seis dígitos cada que fica abaixo da imagem.

Figura 7: EAN.



Fonte: Shopify [4], 2021.

Todavia, com base na imagem acima vemos que a leitura do código obedece uma outra divisão que é feita em quatro partes.

I. Código do País:
   Os três primeiros dígitos indicam qual é o país de origem da mercadoria, o Brasil possui os códigos 789 e 790.

II. Código da empresa:
   Os cinco dígitos a seguir mostram o código do fabricante.

III. Código do produto:

Os quatro próximos códigos são para identificar o produto que está sendo comprado.

IV. Dígito verificador:
É adicionado no final do processo de criação do código, obedecendo a um método que mostraremos a seguir.

2.1 Cálculo do Dígito Verificador

O dígito verificador é uma forma de identificar erros na composição do Código de Barras, é dada a cada produto um dígito verificador, o último dígito da sequência, que retorna se aquele código foi criado de maneira correta.

Vamos denotar cada dígito da sequência por $\alpha_1, \alpha_2, \alpha_3, \dots, \alpha_{11}, \alpha_{12}$ e o digito verificador como β.

Escrevamos a sequência como um vetor

- $x = (\alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5, \alpha_6, \alpha_7, \alpha_8, \alpha_9, \alpha_{10}, \alpha_{11}, \alpha_{12}, \beta)$

E um outro vetor, chamado vetor de pesos da seguinte forma

- $y = (1, 3, 1, 3, 1, 3, 1, 3, 1, 3, 1, 3, 1)$

Calcula-se, agora, o produto escalar dos dois vetores multiplicando cada dígito do primeiro vetor pelo algarismo que ocupa a mesma posição no vetor de pesos.

$$x \cdot y = (\alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5, \alpha_6, \alpha_7, \alpha_8, \alpha_9, \alpha_{10}, \alpha_{11}, \alpha_{12}, \beta) \cdot (1,3,1,3,1,3,1,3,1,3,1,3,1)$$

$$x \cdot y = \alpha_1 + 3\alpha_2 + \alpha_3 + 3\alpha_4 + \alpha_5 + 3\alpha_6 + \alpha_7 + 3\alpha_8 + \alpha_9 + 3\alpha_{10} + \alpha_{11} + 3\alpha_{12} + \beta$$

O valor de β será escolhido de tal forma que a soma acima seja um múltiplo de 10, por isso que o algarismo 3 foi escolhido, já que é o menor algarismo em que ele e o 10 são primos entre si, ou seja, $(3,10) = 1$.

Como exemplo, vamos calcular o dígito verificador do código da figura 7 utilizando o método mostrado anteriormente.

Seja o vetor de informações $x = (7,8,9,9,9,9,9,9,1,2,3,4,\beta)$ e o vetor de pesos $y = (1,3,1,3,1,3,1,3,1,3,1,3,1)$, vamos agora calcular o produto escalar dos vetores.

$$x \cdot y = (7,8,9,9,9,9,9,9,1,2,3,4,\beta) \cdot (1,3,1,3,1,3,1,3,1,3,1,3,1)$$

$$x \cdot y = (7 + 8 \cdot 3 + 9 + 9 \cdot 3 + 9 + 9 \cdot 3 + 9 + 9 \cdot 3 + 1 + 2 \cdot 3 + 3 + 4 \cdot 3 + \beta)$$

$$x \cdot y = 7 + 24 + 9 + 27 + 9 + 27 + 9 + 27 + 1 + 6 + 3 + 12 + \beta$$

$$x \cdot y = 161 + \beta$$

Como o próximo múltiplo de 10 após 161 é o 170, então temos que $\beta = 170 - 161 = 9$. Com isso temos o último dígito do Código de Barras.

2.2 O que significam as diferentes barras (Código UPC)

Ao analisarmos um código de barras comum vemos, facilmente, que existem barras brancas e pretas e, além disso, que existe uma diferença entre as espessuras das barras. Isto se dá porque cada barra possui uma determinada cor e espessura para identificação, conforme a tabela a seguir.

Tabela 1: Valores das listras.

Listras	Valor
Branca Fina	0
Branca Média	00
Branca Grossa	000
Branca Muito Grossa	0000
Preta Fina	1
Preta Média	11
Preta Grossa	111
Preta Muito Grossa	1111

Desta forma, segundo Polcino (RPM 65), as quatro primeiras listras do código a seguir, descontadas as duas primeiras listras que servem como limite para início, seriam lidas da seguinte forma.

Figura 8: Exemplo da tabela.

Fonte: rpm 65 [5], 2021.

A sequência é branca grossa, preta média, branca fina e preta fina o que nos daria em número a sequência 0001101. A partir daí o programa leitor utiliza uma tabela para a conversão da sequência em um número.

Porém, existem um problema a ser solucionado. Como a leitora vai saber se o código está sendo lido da esquerda para a direita ou da direita para a esquerda, ou seja, se a imagem está na posição correta ou se está de cabeça para baixo?

A resposta é bem simples, os dígitos são codificados de duas maneiras diferentes, uma para cada posição de leitura como mostra a tabela a seguir.

Tabela 2: Codificações 1.

Dígito	Do lado esquerdo	Do lado direito
0	0001101	1110010
1	0011001	1100110
2	0010011	1101100
3	0111101	1000010
4	0100011	1011100
5	0110001	1001110
6	0101111	1010000
7	0111011	1000100
8	0110111	1001000
9	0001011	1110100

A codificação dos números da coluna “do lado direito” da tabela acima é obtida trocando 0 por 1 e vice-versa na codificação dos números da coluna “do lado esquerdo”. O mais interessante é que a sequência “do lado esquerdo” possui uma quantidade ímpar de dígitos 1 enquanto a outra sequência possui uma quantidade par, e assim o programa leitor consegue identificar o lado que a imagem do código se encontra posicionada.

2.3 O que significam as diferentes barras (Código EAN)

Com a chegada do código EAN os desenvolvedores esbarraram em um problema complicado, adicionar um dígito ao código de tal forma que a leitora pudesse ler os códigos UPC e EAN igualmente. Como foi dito anteriormente, este novo código traria a identificação do país de origem e, por isso, a necessidade do novo dígito.

Figura 9: EAN-13 e UPC-A.

Fonte: rpm 65 [5], 2021.

Note que, no código EAN-13 da imagem acima, a sequência começa com um zero a mais que o código UPC, porém as imagens para o código leitor são totalmente iguais.

> O dígito inicial 0, a mais no número que nós humanos podemos ler, se explica pelo fato de que os países que utilizavam o antigo código UPC, Estados Unidos e Canadá, foram identificados colocando-se precisamente um 0, na frente, e mantendo o resto da codificação como no sistema anterior. (POLCINO MILIES, RPM 65.)

Com a necessidade de adicionar um dígito ao código e ainda assim manter o tamanho do código utilizado na imagem e também com o intuito de não precisar modificar as máquinas leitoras, a ideia foi que o novo dígito estivesse implícito na escrita de todos os outros, ou seja, a imagem não se alteraria.

Esse problema foi solucionado apenas modificando a codificação do lado esquerdo da imagem por meio de alguns padrões de paridade, que dependem do primeiro dígito que aparecer. Já a codificação do lado direito foi mantida da mesma forma com uma quantidade par de dígitos 1, desta forma as máquinas de leitura continuariam identificando o lado de leitura.

Já na parte esquerda do código a esquematização ficou da seguinte forma, se a sequência começar pelo dígito zero então segue-se o padrão do sistema UPC com uma quantidade ímpar de posições do dígito 1, mas se o primeiro dígito for 1 então os dígitos do lado esquerdo seguirão o seguinte padrão: ímpar, ímpar, par, ímpar, par, par.

Tabela 3: Codificações 2.

Dígito	Lado esquerdo ímpar	Lado esquerdo par	Lado direito
0	0001101	0100111	1110010
1	0011001	0110011	1100110
2	0010011	0011011	1101100
3	0111101	0100001	1000010
4	0100011	0011101	1011100
5	0110001	0111001	1001110
6	0101111	0000101	1010000
7	0111011	0010001	1000100
8	0110111	0001001	1001000
9	0001011	0010111	1110100

Após a escolha do dígito inicial teremos uma alternância diferente de pares e ímpares seguindo o critério da próxima tabela.

Tabela 4: Paridade.

Dígito inicial	1º	2º	3º	4º	5º	6º
0	ímpar	ímpar	ímpar	ímpar	ímpar	Ímpar
1	ímpar	ímpar	par	Ímpar	par	par
2	ímpar	ímpar	par	par	ímpar	par
3	ímpar	ímpar	par	par	par	Ímpar
4	ímpar	par	ímpar	Ímpar	par	par
5	ímpar	par	par	ímpar	Ímpar	par
6	ímpar	par	par	par	ímpar	ímpar
7	ímpar	par	ímpar	par	ímpar	par
8	ímpar	par	ímpar	par	par	ímpar
9	ímpar	par	par	ímpar	par	ímpar

Vamos agora para um exemplo de código de barras de um refrigerante produzido no Brasil, que é registrado pelo código 7894900093308. Como está registrado no Brasil como país fabricante, o código começa com 789 e, sendo assim, o dígito sete, que é o primeiro, terá sua escrita implícita nos demais dígitos.

Após analisarmos a tabela anterior temos que a codificação do lado esquerdo obedecerá à seguinte ordem: ímpar, par, ímpar, par, ímpar, par.

Consultando a tabela de codificações EAN-13 obtemos o seguinte:

$$8 \rightarrow 0110111$$

$$9 \rightarrow 0010111$$

$$4 \rightarrow 0100011$$
$$9 \rightarrow 0010111$$
$$0 \rightarrow 0001101$$
$$0 \rightarrow 0100111$$

Temos então que os dígitos 8, 4 e 0 (primeiro zero) terão uma quantidade ímpar de 1 e os dígitos 9, 9 e 0 (segundo zero) terão uma quantidade par de 1.

Agora para o lado direito temos apenas uma tabela de codificação, e então, não há necessidade de uma tabela auxiliar para os últimos seis dígitos do código. Logo, a codificação ficará da seguinte forma:

$$0 \rightarrow 1110010$$
$$9 \rightarrow 1110100$$
$$3 \rightarrow 1000010$$
$$3 \rightarrow 1000010$$
$$0 \rightarrow 1110010$$
$$8 \rightarrow 1001000$$

E, finalmente, temos a imagem do código de barras que representará esse produto.

Figura 10: Código de barras de refrigerante.

Fonte: autor.

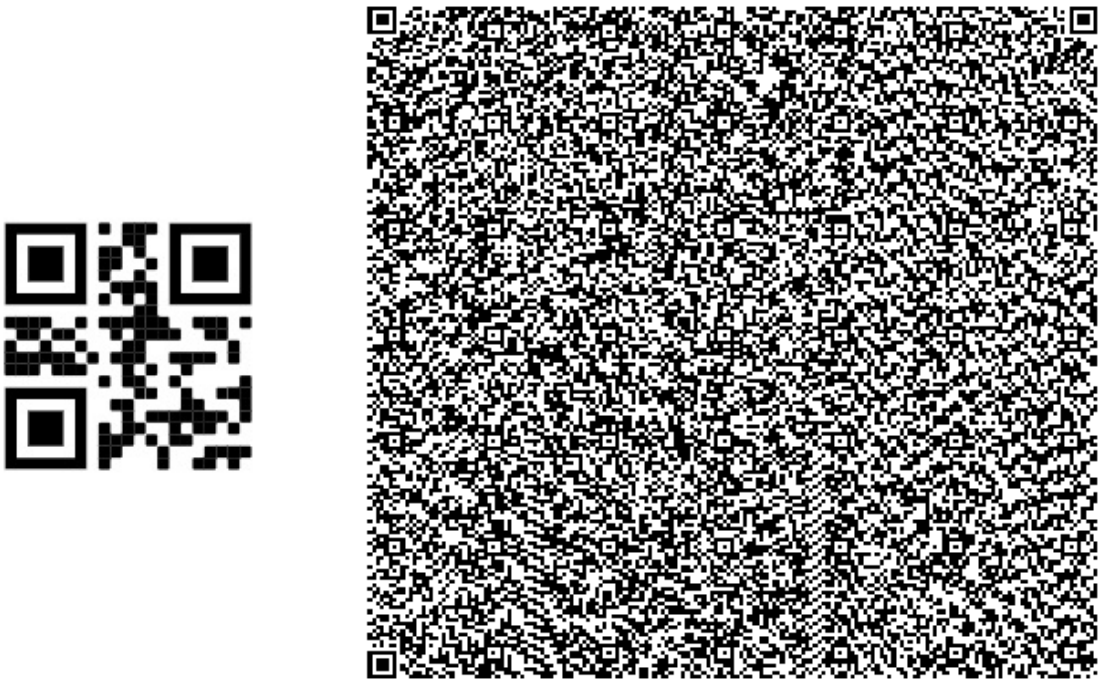
3. COMO FUNCIONA O QR CODE

Neste trabalho utilizaremos como modelo de estudo o QR Code Model 2 Version 1, embora haja outras variações do QR code, como o iQR Code e o Micro QR Code. Neste capítulo vamos mostrar suas subdivisões, quais os seus elementos e funções. Pode parecer à primeira vista ser de difícil compreensão a mecânica envolvida na sua construção, mas iremos mostrar passo a passo cada detalhe envolvido neste código mensageiro.

3.1 Quantidade de pixels

O QR Code é um quadrado subdividido em pequenos quadradinhos chamados pixels, que vão nos dar uma noção da capacidade de armazenagem de informação que a imagem terá. Um código é separado em versões e possui de $21 \times 21 = 441$ pixels na versão 1 até $177 \times 177 = 31329$ pixels na versão 40 (versão máxima).

Figura 11: Versão 1 e Versão 40.



Fonte: autor.

Acima podemos ver a diferença de tamanho entre a menor e a maior versão do código QR e abaixo temos uma tabela com todas as versões do QR e seus respectivos

pixels (linha x coluna). Note que a quantidade de pixels por versão cresce de quatro em quatro.

Tabela 5: quantidade de pixels.

Versão	Pixels	Versão	Pixels	Versão	Pixels	Versão	Pixels
1	21x21	11	61x61	21	101x101	31	141x141
2	25x25	12	65x65	22	105x105	32	145x145
3	29x29	13	69x69	23	109x109	33	149x149
4	33x33	14	73x73	24	113x113	34	153x153
5	37x37	15	77x77	25	117x117	35	157x157
6	41x41	16	81x81	26	121x121	36	161x161
7	45x45	17	85x85	27	125x125	37	165x165
8	49x49	18	89x89	28	129x129	38	169x169
9	53x53	19	93x93	29	133x133	39	173x173
10	57x57	20	97x97	30	137x137	40	177x177

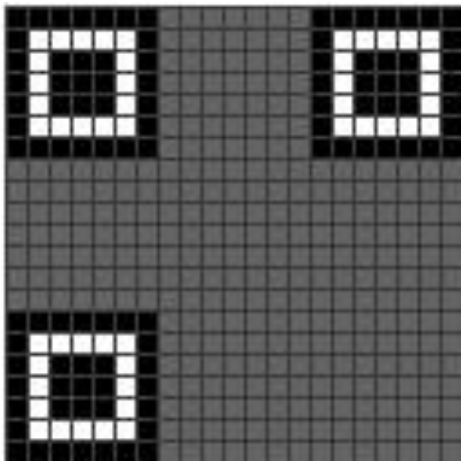
Fonte: autor.

Neste trabalho utilizaremos a versão 21x21 do QR Code como base para a maioria dos exemplos, já que é a menor delas e mais simples e com isso facilitará a explicação e o entendimento dos leitores.

3.2 Padrão de localização

Uma forma de identificar um QR Code é olhando os padrões que ficam em três dos quatro cantos do quadrado, esses padrões distintivos possuem o mesmo tamanho e aparecem em qualquer tipo de QR Code para detectar a posição de rotação da imagem.

Figura 12: Padrão de leitura

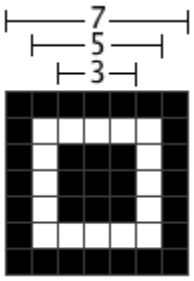


Fonte: Thonky [6], 2021.

Esta é a posição padrão do QR Code, um padrão no vértice esquerdo inferior, um padrão no vértice esquerdo superior e um padrão no vértice direito superior, com isso, o leitor de código, por exemplo a câmera do celular, consegue fazer a leitura mesmo se a imagem estiver rotacionada. Outra função do padrão é determinar o tamanho do pixel. Cada padrão é formado por 7×7 pixels, ou seja, são 49 pixels divididos em 3 grupos.

- Grupo preto externo com 24 pixels;
- Grupo branco com 16 pixels;
- Grupo preto interno com 9 pixels.

Figura 13: Pixels do padrão.



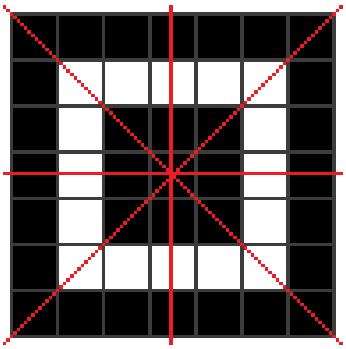
Fonte: Thonky [6], 2021.

Segundo Thonky [12], o padrão de localização foi feito para ser um padrão que provavelmente não aparecerá nas outras seções do QR Code. As larguras dos módulos do padrão do localizador têm uma proporção de 1: 1: 3: 1: 1. Os leitores de QR Code podem pesquisar essa proporção de módulos claros para escuros para detectar os padrões do localizador e orientar corretamente o QR Code para decodificação.

A imagem abaixo mostra a proporção citada, com as linhas vermelhas mostrando a seguinte sequência de pixels em qualquer direção:

- 1 preto
- 1 branco
- 3 pretos
- 1 branco
- 1 preto

Figura 14: Padrão 1:1:3:1:1.

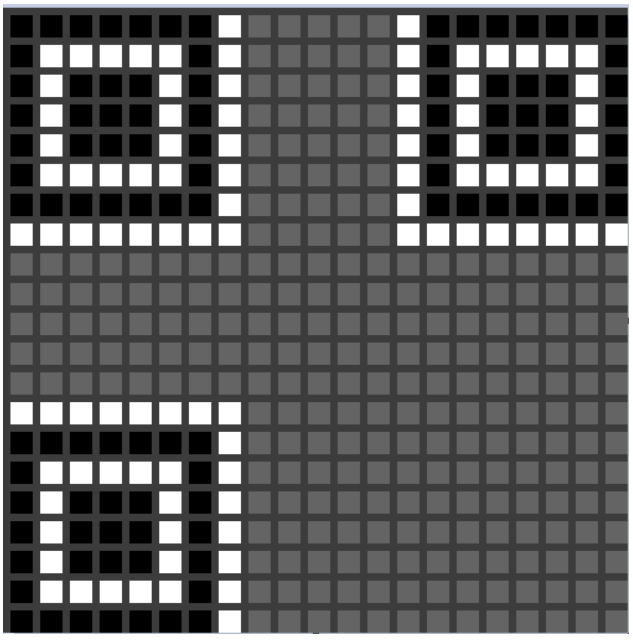


Fonte: autor

3.3 Separadores

Os chamados separadores são linhas brancas que são colocadas ao lado dos padrões distintivos para separá-los do restante do QR Code.

Figura 15: Separadores.



Fonte: Thonky [6], 2021.

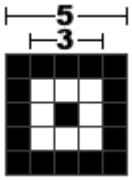
A função deste elemento do código é separar o padrão do restante do código, dando destaque aos três padrões citados na seção anterior e evitar erros de leitura. São compostas de três partes em formatos de L e cada parte é formada por 15 pixels brancos.

3.4 Padrão de alinhamento

QR Codes cuja imagem é muito grande precisam de um padrão de alinhamento para orientar o programa a fazer a leitura. Esse padrão é composto por um quadrado de tamanho $5 \times 5 = 25$ pixels também subdividido em três grupos.

- Grupo preto externo com 16 pixels;
- Grupo branco com 8 pixels;
- Grupo preto interno com 1 pixel.

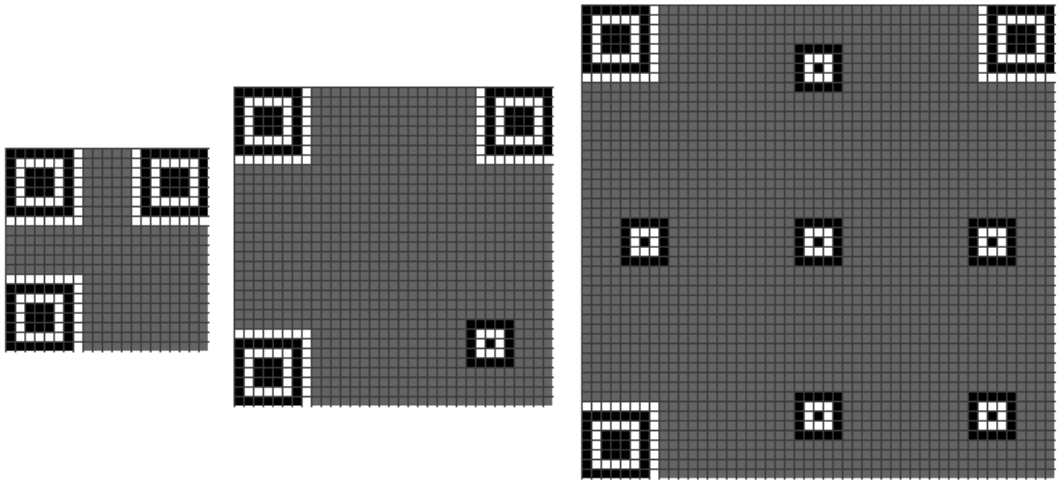
Figura 16: Estrutura de alinhamento.



Fonte: Thonky [6], 2021.

Na versão 1 (21×21), que é o foco deste texto, esse elemento da estrutura do código não aparece, conforme abaixo.

Figura 17: Estrutura de alinhamento no QR Code.



Fonte: autor.

Na imagem acima temos três versões do QR Code que são versão 1 (21x21), versão 4 (33x33) e versão 8 (49x49), note que quanto maior for a versão do QR Code maior será a quantidade de estruturas de alinhamento para orientar o escâner.

Este padrão possui uma localização específica dentro de cada QR Code dependendo de sua versão, vide tabela a seguir.

Tabela 6: coordenadas dos padrões de alinhamento.

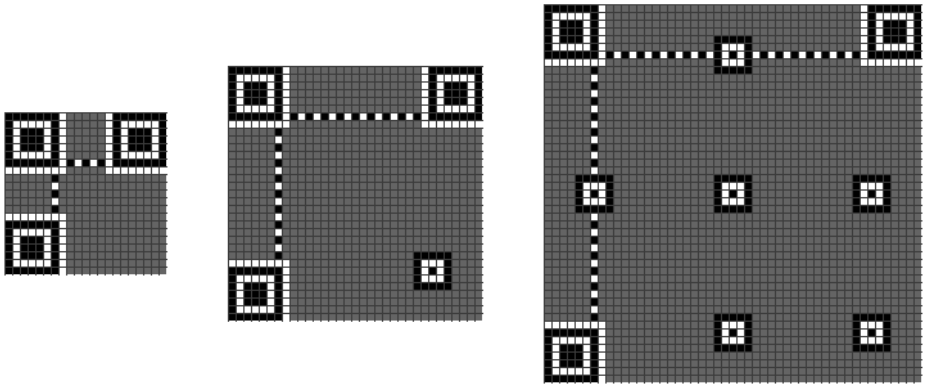
Versão	Linha e coluna				Versão	Linha e coluna						
QR Versão 2	6	18			QR Versão 21	6	28	50	72	94		
QR Versão 3	6	22			QR Versão 22	6	26	50	74	98		
QR Versão 4	6	26			QR Versão 23	6	30	54	78	102		
QR Versão 5	6	30			QR Versão 24	6	28	54	80	106		
QR Versão 6	6	34			QR Versão 25	6	32	58	84	110		
QR Versão 7	6	22	38		QR Versão 26	6	30	58	86	114		
QR Versão 8	6	24	42		QR Versão 27	6	34	62	90	118		
QR Versão 9	6	26	46		QR Versão 28	6	26	50	74	98	122	
QR Versão 10	6	28	50		QR Versão 29	6	30	54	78	102	126	
QR Versão 11	6	30	54		QR Versão 30	6	26	52	78	104	130	
QR Versão 12	6	32	58		QR Versão 31	6	30	56	82	108	134	
QR Versão 13	6	34	62		QR Versão 32	6	34	60	86	112	138	
QR Versão 14	6	26	46	66	QR Versão 33	6	30	58	86	114	142	
QR Versão 15	6	26	48	70	QR Versão 34	6	34	62	90	118	146	
QR Versão 16	6	26	50	74	QR Versão 35	6	30	54	78	102	126	150
QR Versão 17	6	30	54	78	QR Versão 36	6	24	50	76	102	128	154
QR Versão 18	6	30	56	82	QR Versão 37	6	28	54	80	106	132	158
QR Versão 19	6	30	58	86	QR Versão 38	6	32	58	84	110	136	162
QR Versão 20	6	34	62	90	QR Versão 39	6	26	54	82	110	138	166
					QR Versão 40	6	30	58	86	114	142	170

Os números no lado direito desta tabela devem ser usados como ambas as coordenadas de linha e coluna. Por exemplo, a versão 2 tem os números 6 e 18. Isso significa que os centros dos padrões de alinhamento devem ser colocados em (6, 6), (6, 18), (18, 6) e (18, 18). A versão 1 não possui este elemento, por isso a sua ausência na tabela, e a versão 40 possui 46 padrões de alinhamento pois sempre são removidos os padrões que se sobrepuserem aos padrões de localização ou separadores.

3.5 Padrões de tempo

Esta parte do código é formada por duas linhas que são feitas de pixels de cores alternadas, sempre começando e terminando por um pixel preto.

Figura 18: Padrão de tempo.



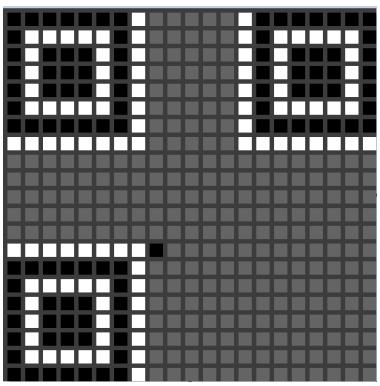
Fonte: Thonky [6], 2021.

Esta estrutura sempre é colocada na sexta coluna da esquerda para a direita e na sexta linha de cima para baixo. Isso permite ao programa leitor confirmar a versão do código assim como o tempo de cada pixel ao longo da imagem além de auxiliar na percepção do posicionamento das linhas e colunas.

3.6 Pixel Preto (Dark Module)

Este único pixel preto é adicionado ao lado do padrão distintivo inferior e não possui nenhuma relevância para os dados armazenados.

Figura 19: Pixel Preto.



Fonte: Thonky [6], 2021.

Após a alocação deste pixel preto o restante do código é destinado para representar a informação contida na imagem. Este pixel é sempre colocado na coordenada ([(4 * V) + 9], 8), onde V é a versão do QR Code.

Portanto, na versão 1 teremos o Dark Module ocupando a posição $(4 \times 1 + 9, 8) = (13, 8)$, que será o pixel da linha 8, de baixo para cima, e coluna 13, da direita para a esquerda.

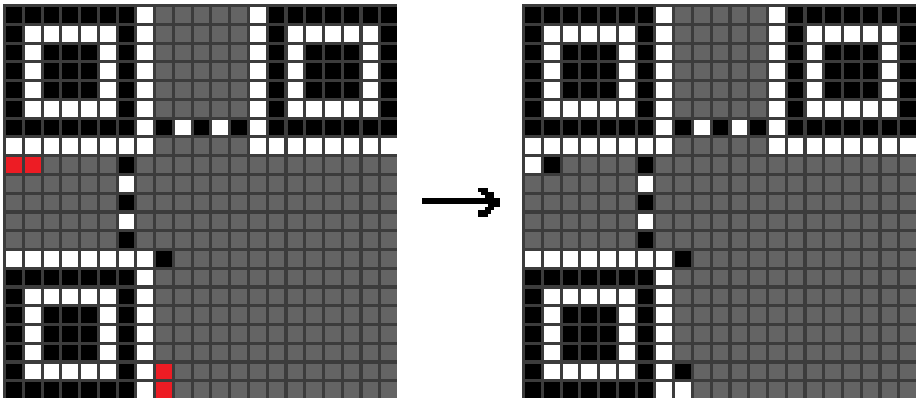
3.7 Correção de erro

O próximo item da estrutura da imagem é onde fica armazenado o nível de correção de erro que será adotado na construção dos dados.

Cada nível de correção de erro reserva uma parte do código para repetição dos dados da imagem, afim de corrigir qualquer falha de leitura da mensagem original ou parte do código que esteja faltando ou danificada.

Este elemento ocupa dois bits da mensagem e fica posicionado na parte vermelha da imagem, como mostrando na figura a seguir.

Figura 20: Posição da correção de erro.



Fonte: autor.

Acima, vemos a posição onde esses pixels dever ser marcados (em vermelho) e, ao lado, vemos um QR Code com o nível L de correção de erro.

São quatro níveis de correção a serem escolhidos (L, M, Q, H).

Tabela 7: Correção de erro.

NÍVEL DE CORREÇÃO DE ERRO	BITS	PIXELS	INTEIRO EQUIVALENTE
L (LOW)	01	■ □	1
M (MEDIUM)	00	□ □	0
Q (QUARTILE)	11	■ ■	3
H (HIGH)	10	□ ■	2

Note, na tabela acima, que os valores associados aos níveis de correção de erro não seguem a ordem dos números inteiros. Os níveis de correção de erro estão ordenados do menor para o maior nível, porém os valores inteiros equivalentes seguem a ordem 1, 0, 3 e 2.

A figura a seguir nos mostra um exemplo de como mesmo a imagem do QR Code sendo danificada ainda será possível ler a mensagem contida nele.

Figura 21: QR Code danificado.

Fonte: Wikipedia [7], 2021.

Veja a tabela abaixo com as porcentagens aproximadas de redundância da mensagem.

Tabela 8: Porcentagem da correção de erro.

$\boldsymbol{L\ (LOW) \approx 7\%}$
$\boldsymbol{M\ (MEDIUM)\ \approx 15\%}$
$\boldsymbol{Q\ (QUARTILE) \approx 25\%}$
$\boldsymbol{H\ (HIGH) \approx 30\%}$

3.8 Tipos de dados armazenados

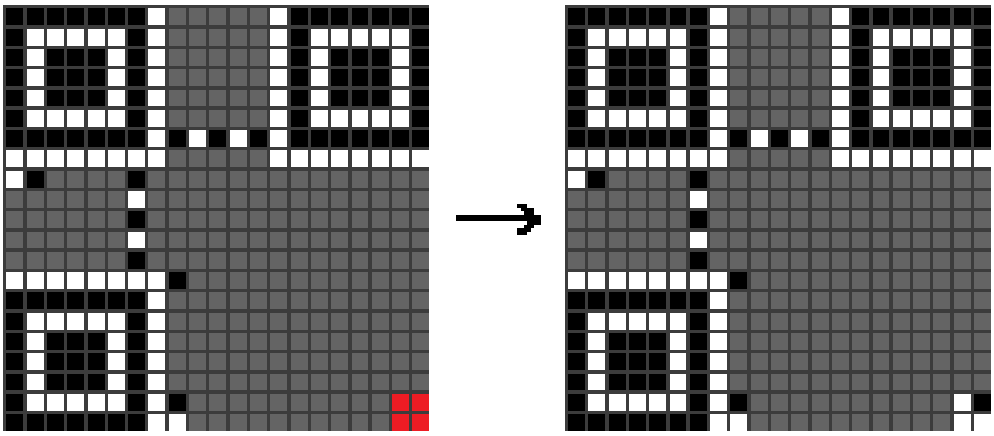
Os quatro primeiros bits da mensagem do código são para identificar o tipo de caracteres contidos na informação, que são: numéricos, alfanuméricos, byte, kanji, ECI, entre outros, além da possibilidade de combinação de tipos. Vamos mostrar os cinco principais abaixo.

Tabela 9: Tipos de dados.

TIPO	BITS	PIXELS
NUMÉRICO	0001	
ALFANUMÉRICO	0010	
BYTE	0100	
KANJI	1000	
ECI	0111	

Esses quatro bits ficam posicionados no canto inferior direito como podemos ver, a seguir, na figura 22.

Figura 22: Posição dos tipos de dados armazenados.



Fonte: autor.

Acima, à esquerda, vemos em vermelho os quatro bits destinados a este elemento e, à direita, vemos um QR Code preenchido com o tipo alfanumérico.

Segundo Thonky [12], os quatro modos de codificação incluem os seguintes caracteres:

O numérico é para dígitos decimais de 0 a 9.

O modo alfanumérico é para os dígitos decimais de 0 a 9, bem como letras maiúsculas e os símbolos $,%, *, +, -,., / e o espaço.

Tabela 10: Codificação alfanumérica.

0	0	F	15	U	30
1	1	G	16	V	31
2	2	H	17	W	32
3	3	I	18	X	33
4	4	J	19	Y	34
5	5	K	20	Z	35
6	6	L	21	Espaço	36
7	7	M	22	$	37
8	8	N	23	%	38
9	9	O	24	*	39
A	10	P	25	+	40
B	11	Q	26	-	41
C	12	R	27	.	42
D	13	S	28	/	43
E	14	T	29	:	44

Na tabela acima, temos nas colunas em branco os caracteres supracitados e nas colunas em cinza os valores que cada caractere assumirá na codificação. Por exemplo, se um byte da mensagem carregar como informação a letra A então na sua codificação aparecerá o número 10. Todas as informações sobre bytes e codificações serão expostas nas seções a seguir.

O modo byte, por padrão, é para caracteres do conjunto de caracteres ISO-8859-1 (codificação de caracteres do alfabeto latino). No entanto, alguns scanners de QR Code podem detectar automaticamente se UTF-8 (tipo de codificação binária para caracteres) é usado no modo byte.

E, finalmente, o modo Kanji (Os kanji são caracteres da língua japonesa adquiridos a partir de caracteres chineses) é para caracteres de byte duplo do conjunto de caracteres Shift JIS, que é uma codificação de caracteres para o idioma japonês, originalmente desenvolvido por uma empresa japonesa chamada ASCII Corporation em conjunto com a Microsoft.

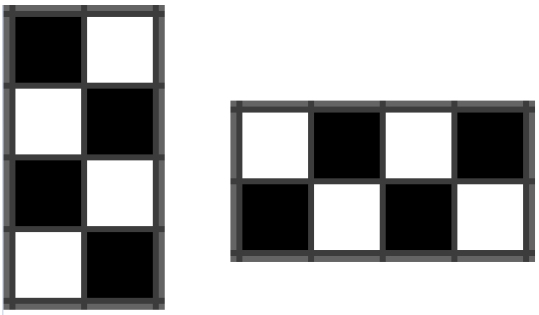
Segundo Thonky, se a mensagem consistir apenas em dígitos decimais (0 a 9), será utilizado o modo numérico, se o modo numérico não for suficiente para cobrir toda a informação então será aplicado o modo alfanumérico conforme a tabela 10, se houver um caractere que não está na tabela anterior, mas pode ser codificado em ISO 8859-1, aplicar-se-á o modo byte ou, caso ainda seja necessário, a mensagem será codificada em modo Kanji.

O modo Byte será decodificado pela tabela ASCII que possui 256 caracteres, e é o modo de codificação mais utilizado. Ele precisa de 8 bits para escrever um caractere.

3.9 Byte

Os bytes são responsáveis por carregar a mensagem contida na imagem do QR Code, são blocos de 8 bits, como mostra a figura abaixo, mas dependendo da versão e tipo de dados armazenados a quantidade de bits em um byte pode mudar. Neste trabalho iremos utilizar o 8-bit-byte, ou seja, os bytes de tamanho 8.

Figura 23: Bytes.



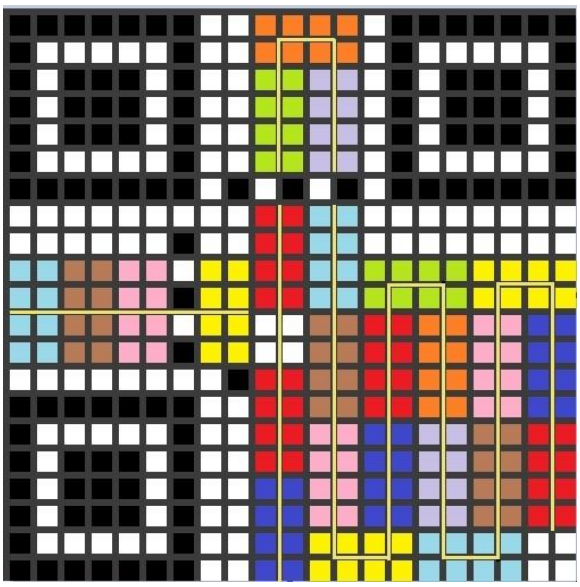
Fonte: autor.

Este elemento do QR Code pode aparecer na posição vertical (4 linhas e 2 colunas) ou horizontal (2 linhas e 4 colunas).

Por padrão, o primeiro byte do QR Code é responsável por carregar a informação sobre a quantidade total de bytes contidos na imagem, ou seja, se um QR Code carrega uma mensagem com 10 caracteres, a palavra “matemática” por exemplo, então o primeiro byte terá o número 11 como informação.

Os bytes da mensagem seguirão o padrão da imagem a seguir, sendo o primeiro byte (em vermelho) o que carrega a informação da quantidade total de bytes, e será alocado logo acima dos quatro primeiros bits no canto inferior direito.

Figura 24: Caminho do byte.



Fonte: autor.

A linha em dourado começa no primeiro byte (vermelho) e segue o caminho da informação dentro do QR Code, ou seja, o segundo byte está na cor azul escuro, o terceiro na cor amarela e assim por diante, até chegar no último byte de cor azul claro.

Veja a seguir a ordem dos bytes dentro da imagem nomeados pelas suas respectivas cores.

Tabela 11: Posições e cores dos bytes

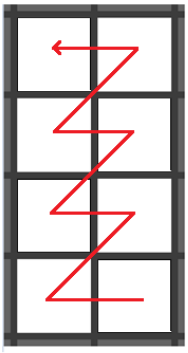
1º	Vermelho	6 º	Azul Claro	11 º	Azul Escuro	16 º	Cinza	21 º	Azul
2 º	Azul Escuro	7 º	Cinza	12 º	Amarelo	17 º	Laranja	22 º	Amarelo
3 º	Amarelo	8 º	Laranja	13 º	Rosa	18 º	Verde	23 º	Rosa
4 º	Rosa	9 º	Verde	14 º	Marrom	19 º	Vermelho	24 º	Marrom
5 º	Marrom	10 º	Vermelho	15 º	Azul Claro	20 º	Vermelho	25 º	Azul Claro

Note que, na figura 24, entre os 19º e 20º bytes existe um bloco com quatro bits brancos. Ele indica o fim da mensagem original, e após esse bloco a mensagem será repetida até que se complete todo o espaço restante do QR Code destinado aos bytes.

Essa repetição é necessária para que o programa leitor possa fazer a restauração de alguma parte danificada do QR Code, como visto, na seção 3.7.

Dentro de cada byte, os bits serão alocados respeitando a ordem mostrada nas figuras a seguir. Caso os bytes estejam sendo alocados no sentido de baixo para cima os bits serão alocados também de baixo para cima.

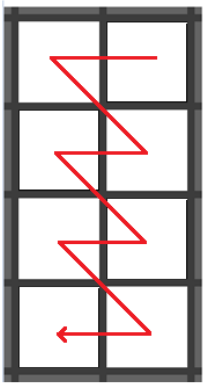
Figura 25: Caminho do bit 1.



Fonte: autor.

Caso os bytes estejam sendo alocados de cima para baixo, os bits também serão alocados neste sentido.

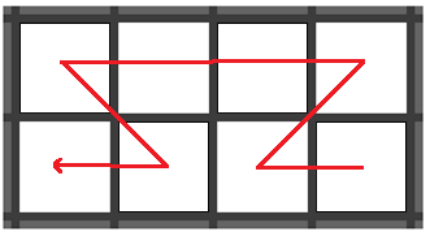
Figura 26: Caminho do bit 4.



Fonte: autor.

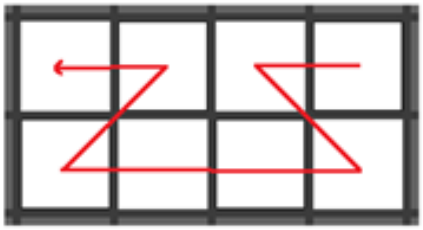
O byte também aparecerá na posição horizontal, com duas linhas e quatro colunas de bits, e será preenchido seguindo as orientações mostradas nas duas próximas figuras.

Figura 27: Caminho do bit 3.



Fonte: autor.

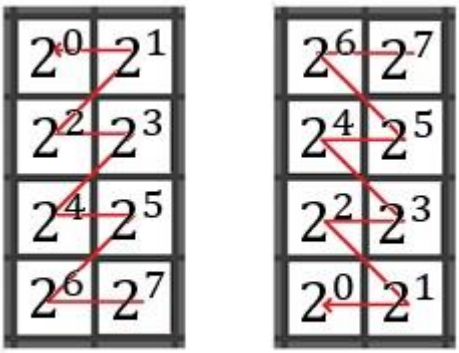
Figura 28: Caminho do bit 6.



Fonte: autor.

Cada bit possuirá um valor diferente dentro dos bytes, sendo o primeiro bit o mais significativo e o último bit o menos significativo. Veja a seguir.

Figura 29: Potências de 2 nos bits.



Fonte: autor.

Exemplo:

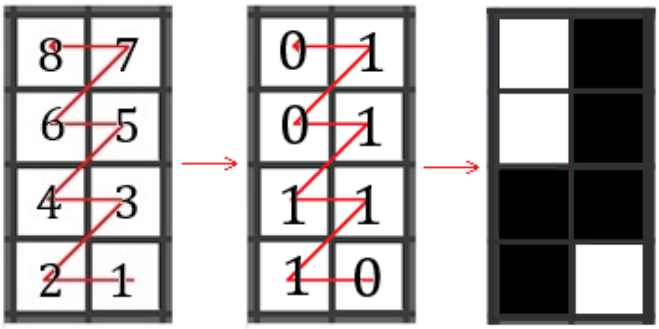
Um byte terá como mensagem a letra “z” que na codificação do Modo Byte terá o valor de 01111010. Se sua escrita é de baixo para cima, como ficará o preenchimento?

Primeiro, para facilitar, organizamos os bits de 1 a 8

1	2	3	4	5	6	7	8
0	1	1	1	1	0	1	0

Vamos agora preencher cada quadradinho da imagem a seguir, de acordo com a numeração da primeira linha da tabela anterior. De acordo com a segunda linha, cada bit que estiver relacionado com o dígito 1 será pintado de preto.

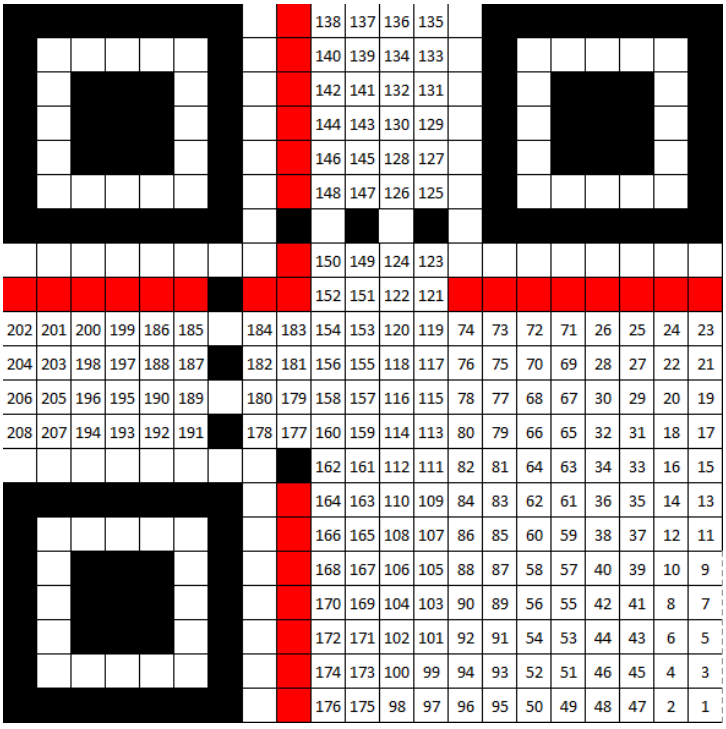
Figura 30: Completando um byte.



Fonte: autor

A seguir, poderemos ter uma visão geral do comportamento de escrita e leitura dos bits na próxima figura

Figura 31: Escrita e leitura dos bits



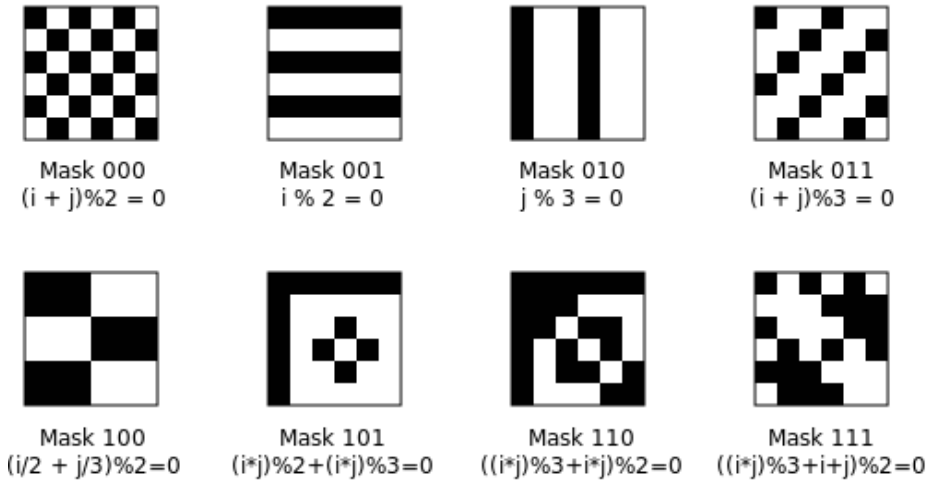
Fonte: autor.

Iniciando pelo bit número 1 na parte inferior direita, que pertence ao bloco dos tipos de dados armazenados, a escrita da informação vai seguindo a ordem dos números naturais até atingir o último bit no número 208. Note que este tipo de escrita segue um padrão de “ziguezague” sempre alternando colunas.

3.10 Máscara

As máscaras têm como objetivo modificar a apresentação final do QR Code a fim de evitar grandes blocos de pixels de uma mesma cor, o que pode ocasionar dificuldades ao aplicativo leitor para escanear a imagem. Ela é uma camada de pixels que se sobrepõem ao código original, apenas a parte da mensagem. Além disso, ela evita a aparição do padrão de alinhamento “1011101” no corpo da mensagem. Abaixo temos os oito tipos de máscaras.

Figura 32: Tipos de máscaras



Fonte: wikimedia [9], 2021.

Cada máscara possui uma numeração que vai de 0 a 7 convertida para binário, como mostrado na imagem anterior.

Cada pixel da máscara tem uma função específica

- Pixel preto inverte a cor do pixel original.
- Pixel branco mantém a cor do pixel original.

Note que abaixo de cada máscara existe uma fórmula modular que indica a operação que será feita para determinar a mudança, ou não, que cada bit da mensagem irá sofrer. Essa fórmula pode ser interpretada de forma fácil como veremos a seguir.

O processo de posicionamento das máscaras segue a ordem de fila i e de coluna j, sendo $(i,j) = (0,0)$ a posição superior esquerda da máscara. Na máscara 000 temos que $(i + j)\%2 = 0$ que significa que a soma do valor da linha i com o valor da coluna j tem que ter divisível por 2, ou seja, tem que deixar resto zero quando dividido por 2.

Exemplo: Um bit que está localizado na posição $(i,j) = (11,7)$ sofrerá mudança de cor ou não?

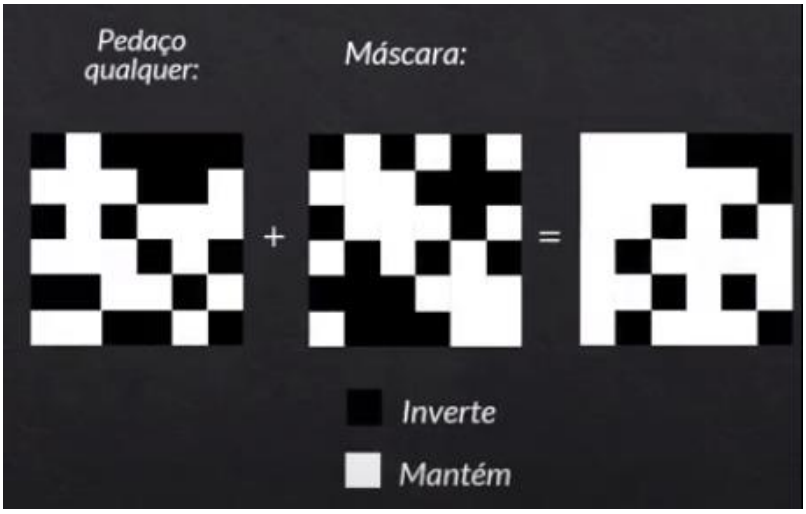
Resposta:

Este bit sofrerá a ação de mudança de valor pois $11 + 7 = 18$ e $18 \div 2 = 9$.

Já na máscara que possui o valor 001 e fórmula $i\%2 = 0$ são os valores das linha i que quando divididos por 2 deixam resto 0, ou seja, as linhas pares. Por este motivo a máscara 001 é formada por linhas pretas e brancas alternadas, sendo que as pretas ocupam as linhas de valor par.

Desta forma, se um pixel branco (ou preto) original do QR Code encontrar um pixel preto da máscara, o pixel mostrado na imagem final será preto (ou branco). Veja abaixo um exemplo.

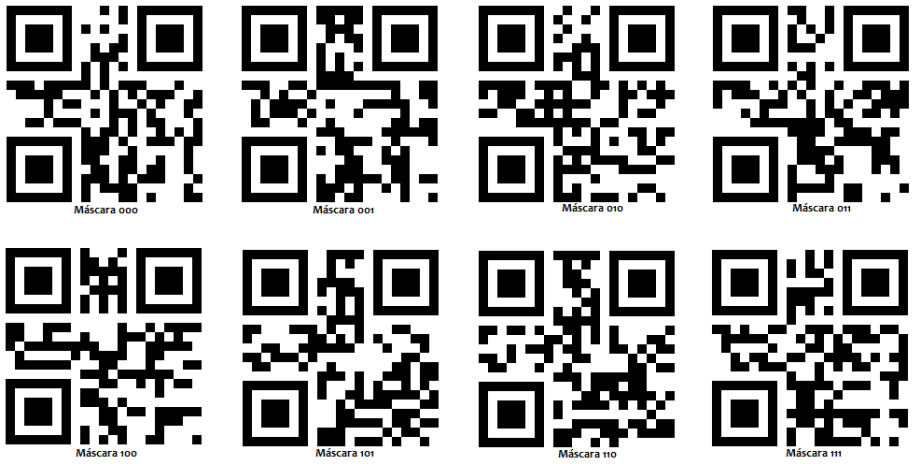
Figura 33: Aplicação da máscara.



Fonte: Youtube [10], 2021.

Veja a seguir uma imagem contendo oito QR Codes onde foi codificada a mensagem “profmat 2021” e cada um foi sobreposto por um tipo de máscara diferente.

Figura 34: profmat 2021.



Fonte: autor.

A partir daí surge uma pergunta, como o programa escolhe a máscara a ser utilizada?

O programa gerador do QR Code selecionará a imagem com a menor penalidade, sendo que as penalidades são definidas de quatro formas para melhor determinar a distribuição entre bits pretos e brancos.

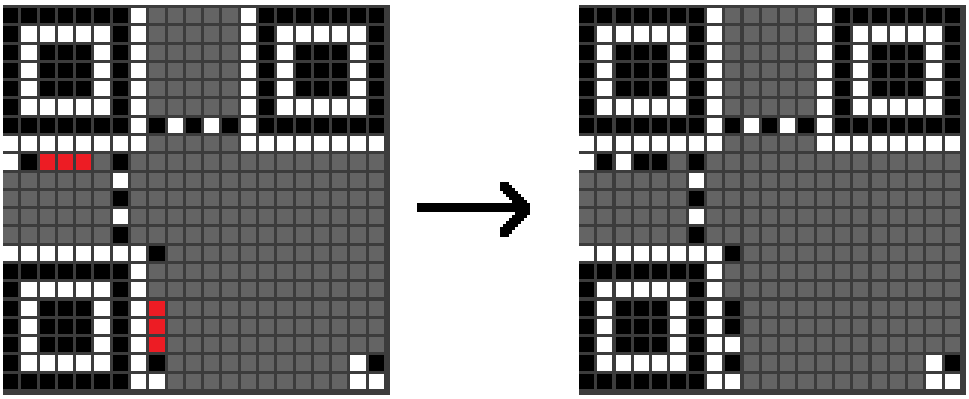
Tabela 12: Penalidades.

Penalidade	Característica	Condição	Pontos
1	Grupo de cinco ou mais bits da mesma cor em uma linha (ou coluna).	Número de bits $= 5 + i$	$3 + i$
2	Área de, pelo menos, 2×2 bits da mesma cor	Tamanho do bloco $= m \times n$	$3 \times (m - 1) \times (n - 1)$
3	Existência da razão 1:1:3:1:1 que tenha quatro bits brancos em cada lado.	Existência do padrão	40
4	Proporção entre bits brancos e pretos	$50 \pm (5 \times k)\%$ a $50 \pm \big(5 \times (k + 1)\big)\%$	$40 \times k$

Na tabela acima k é o desvio da proporção de bits pretos em torno dos 50% esperados em passos de 5% e i é a quantidade de bits adjacentes de mesma cor que excede o valor 5.

Na figura 34 a máscara escolhida pelo programa foi a de valor 100 por possuir a menor penalidade entre todas. Cada máscara tem uma numeração de 3 dígitos e essa numeração ocupará os bits em vermelho na figura a seguir.

Figura 35: Numeração da máscara.



Fonte: autor.

Acima temos um QR Code com a máscara 011, ou seja, bits branco-preto-preto. A máscara é a última etapa a ser adicionada ao QR Code.

Após todos esses passos, o programa criador do QR Code irá, através de divisões polinomiais, codificar a informação de formato que terá 15 dígitos. A string de formato sempre tem 15 bits de comprimento.

> Para criar a string, primeiro você cria uma string de cinco bits que codifica o nível de correção de erros e o padrão de máscara em uso neste código QR. Em seguida, você usa esses cinco bits para gerar dez bits de correção de erro. Os quinze bits resultantes são XORed. (Thonky [11], 2021)

Abaixo temos a tabela completa com as 32 codificações possíveis através dos 4 tipos de nível de correção de erro (low, medium, quartile e high) e dos 8 tipos de máscaras possíveis numeradas de 0 até 7.

Tabela 13: Lista de todas as strings de informações de formato

Nível de correção de erro	Padrão da Máscara	Tipo de bits de informação	Nível de correção de erro	Padrão da Máscara	Tipo de bits de informação
L	0	111011111000100	Q	0	011010101011111
L	1	111001011110011	Q	1	011000001101000
L	2	111110110101010	Q	2	011111100110001
L	3	111100010011101	Q	3	011101000000110
L	4	110011000101111	Q	4	010010010110100
L	5	110001100011000	Q	5	010000110000011
L	6	110110001000001	Q	6	010111011011010
L	7	110100101110110	Q	7	010101111101101
M	0	101010000010010	H	0	001011010001001
M	1	101000100100101	H	1	001001110111110
M	2	101111001111100	H	2	001110011100111
M	3	101101101001011	H	3	001100111010000

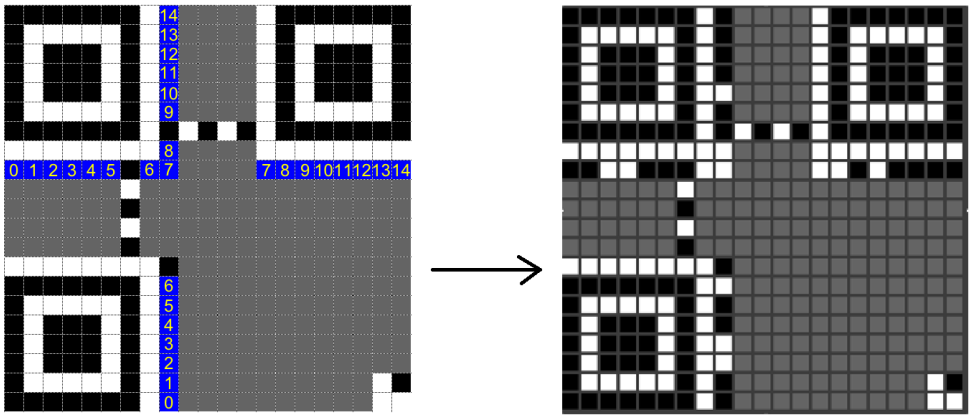
M	4	100010111111001	H	4	000011101100010
M	5	100000011001110	H	5	000001001010101
M	6	100111110010111	H	6	000110100001100
M	7	100101010100000	H	7	000100000111011

Exemplo: Um QR Code com nível de correção de erro L e padrão de máscara 4 terá o número 110011000101111 como informação de formato de acordo com a tabela anterior. Numerando cada dígito, da esquerda para a direita, de 0 a 14 temos:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	1	0	0	1	1	0	0	0	1	0	1	1	1	1

Na imagem a seguir poderemos ver o resultado final após o processo de escolha da correção de erro e da máscara e, enfim, a codificação do formato e o preenchimento de cada bit em azul de acordo com sua numeração. Note que os bits de 0 a 4 anteriormente preenchidos com o nível de correção e tipo de máscara serão sobrepostos pela nova numeração.

Figura 36: Codificação do formato.



Fonte: Thonky [11], 2021.

3.11 Aplicativos e sites de QR Code

Os novos smartphones já vem de fábrica com leitor de QR Code embutido na câmera do aparelho, porém, muitos ainda não possuem esta funcionalidade e, com isso, os usuários podem baixar alguns aplicativos gratuitos para a leitura. Veja abaixo alguns exemplos de apps.

Figura 37: Aplicativos de leitura de QR Code e Código de Barras.

Fonte: autor.

Esses aplicativos possuem as funções de criar e ler os códigos, além disso, alguns deles podem inserir logotipos nos QR Codes e muito mais. Alguns sites também disponibilizam essas funções gratuitamente.

- https://www.qrcodefacil.com/
- https://br.qr-code-generator.com/
- https://www.flowcode.com/
- https://br.qr-code-generator.com/

Outros sites disponibilizam até funções para fins de estudo onde é possível clicar na opção “Show Advanced Options” e escolher o tipo de máscara.

- https://www.thonky.com/qrcode/

Para criar um QR Code usando um aplicativo é fácil. Vamos utilizar o aplicativo QRTiger como exemplo. Na primeira imagem temos a página principal do aplicativo, clicando em “Texto” abre-se a segunda tela onde digitamos a mensagem desejada e clicamos em “GERAR CÓDIGO QR”. Na terceira imagem há opções de personalização do código, como cores, formatos do módulo, inserção de logotipo.

Figura 38: Criando um QR Code.

Fonte: autor.

Finalmente, clicamos em “Baixe gratuitamente” e a imagem será salva na galeria de fotos do celular. Veja abaixo um exemplo de QR Code personalizado feito como o passo a passo anterior.

Este possui personalizações como padrão de alinhamento, pixels em formatos diferentes e coloridos, e logotipo no centro.

4. ATIVIDADES EM SALA DE AULA

Neste capítulo vamos mostrar algumas atividades sugeridas para aplicação em sala de aula. Alguns dos exercícios são dos principais vestibulares nacionais e outros são de criação própria do autor sobre algumas características dos códigos que possam ser exploradas em uma turma.

Atividade 1

(COPEVE-UFAL - 2014 - Prefeitura de Feira Grande - AL - Professor - 1º ao 5º ano)

O código de barras de identificação dos equipamentos de uma empresa é formado por uma sequência de 3 barras de 2,5 mm e 4 barras de 1,0 mm. Se cada sequência identifica um único equipamento, quantos equipamentos diferentes podem ser identificados?

a) 11
b) 12
c) 30
d) 32
e) 35

Resolução:

Cada sequência é constituída por um total de 7 barras, sendo 3 barras de 2,5 mm e 4 barras de 1,0 mm.

Logo, teremos uma sequência de 7 elementos, sendo 3 elementos são iguais entre si e os outros 4 elementos são iguais entre si. Assim há uma permutação com repetição.

Então temos que

$$P_7^{4,3} = \frac{7!}{4!\,3!} = \frac{7 \times 6 \times 5 \times 4!}{4! \times 3 \times 2 \times 1} = \frac{7 \times 6 \times 5}{3 \times 2 \times 1} = 35$$

Desta forma, vemos que a empresa consegue cadastrar até 35 equipamentos utilizando este modelo de código de barras.

Atividade 2

(UTFPR - 2018 - UTFPR - Assistente em Administração)

Um código de barras UPC, que em inglês significa Universal Product Code, é uma representação de uma sequência de 13 algarismos na forma gráfica (barras). As barras são reconhecidas por aparelhos decodificadores e identificam o produto. A sequência numérica correspondente às barras também pode ser digitada manualmente. O último algarismo do código é o dígito verificador. Para encontrar esse dígito, realiza-se um cálculo com os 12 algarismos anteriores, da seguinte forma:

1) Somam-se os algarismos das posições ímpares;
2) Somam-se os algarismos das posições pares e multiplica-se o resultado por 3;
3) Somam-se os resultados das etapas anteriores (1) e (2);
4) O dígito verificador será o algarismo que adicionado ao resultado da etapa (3) gere um múltiplo de 10.

Considerando código de barras: 502523265897X, assinale a alternativa que apresenta o dígito verificador X desse código.

a) 5
b) 6
c) 7
d) 8
e) 9

Resolução:

Considerando o código dado que é 502523265897X, vamos aplicar a condição 1 dada.

$$5 + 2 + +2 + 2 + 5 + 9 = 25$$

Agora, apliquemos a condição dada em 2.

$$0 + 5 + 3 + 6 + 8 + 7 = 29$$

Multiplicando o resultado por 3, temos

$$3 \times 29 = 87$$

Na etapa 4 vamos somar os dois resultados anteriores e calcular o valor que falta para o próximo múltiplo de 10.

$$25 + 87 = 112$$

Logo, como o valor achado foi o 112, o próximo múltiplo de 10 que segue é o 120. Sendo assim, temos $X = 120 - 112 = 8$.

Atividade 3

(Vestibular UERJ 2015 - 1ª fase Exame de Qualificação - Adaptado)

Utilize as informações a seguir para responder às atividades 3.1 e 3.2.

Uma loja identifica seus produtos com um código que utiliza 16 barras, finas ou grossas. Nesse sistema de codificação, a barra fina representa o zero e a grossa o 1. A conversão do código em algarismos do número correspondente a cada produto deve ser feita de acordo com esta tabela:

Código	Algarismo	Código	Algarismo
0000	0	0101	5
0001	1	0110	6
0010	2	0111	7
0011	3	1000	8
0100	4	1001	9

Observe um exemplo de código e de seu número correspondente:

Atividade 3.1. Considere o código abaixo, que identifica determinado produto.

Esse código corresponde ao seguinte número:

a) 6835
b) 5724
c) 8645
d) 9768

Resolução:

Como as barras finas possuem o valor de 0 (zero) e as barras pretas possuem o valor de 1 (um), e apenas as barras pretas são utilizadas, temos que a decodificação da imagem seria a seguinte:

$$0110100000110101$$

Vamos agora dividir a sequência em grupos de quatro algarismos para determinar a numeração do produto.

$$0110 - 1000 - 0011 - 0101$$

Analisando a tabela dada e fazendo a conversão dos valores temos que o código de barras possui o valor de 6835.

Atividade 3.2. Existe um conjunto de todas as sequências de 16 barras finas ou grossas que podem ser representadas. Escolhendo-se ao acaso uma dessas sequências, a probabilidade de ela configurar um código do sistema descrito é:

a) $\frac{5}{2^{15}}$

b) $\frac{25}{2^{14}}$

c) $\frac{125}{2^{13}}$

d) $\frac{625}{2^{12}}$

Resolução:

O modelo de código de barras utilizado por essa loja permite o cadastramento de um total de 10.000 produtos, note que cada produto possui um código de quatro algarismos de zero a nove, ou seja, os códigos vão de 0000 a 9999.

O cálculo da quantidade de sequências é da seguinte forma

$$10 \times 10 \times 10 \times 10 = 10.000$$

Agora, tendo em vista que o código gráfico possui 16 barras pretas que podem ser de dois tipos, finas ou grossas, então para cada posição das barras temos duas opções de escolha para preencher, segue que o total de tais sequências de barras é

$$\underbrace{2 \times 2 \times 2 \times 2 \times 2 \times 2 \times 2 \times 2 \times 2 \times 2 \times 2 \times 2 \times 2 \times 2 \times 2 \times 2}_{16\ vezes} = 2^{16}$$

Logo, a probabilidade P de ela configurar um código do sistema descrito é

$$P = \frac{10 \times 10 \times 10 \times 10}{2^{16}} = \frac{5 \times 5 \times 5 \times 5}{2^{12}} = \frac{625}{2^{12}}$$

Atividade 4

(ENEM 2002 - QUESTÃO 27)

O código de barras, contido na maior parte dos produtos industrializados, consiste num conjunto de várias barras que podem estar preenchidas com cor escura ou não. Quando um leitor óptico passa sobre essas barras, a leitura de uma barra clara é convertida no número 0 e a de uma barra escura, no número 1.

Observe abaixo um exemplo simplificado de um código em um sistema de código com 20 barras.

Se o leitor óptico for passado da esquerda para a direita irá ler: 01011010111010110001.

Se o leitor óptico for passado da direita para a esquerda irá ler: 10001101011101011010.

No sistema de código de barras, para se organizar o processo de leitura óptica de cada código, deve-se levar em consideração que alguns códigos podem ter leitura da esquerda para a direita igual à da direita para a esquerda, como o código 00000000111100000000, no sistema descrito acima.

Em um sistema de código de barras que utilize apenas cinco barras, a quantidade de códigos com leitura da esquerda para a direita igual à da direita para a esquerda, desconsiderando-se todas as barras claras ou todas as escuras, é

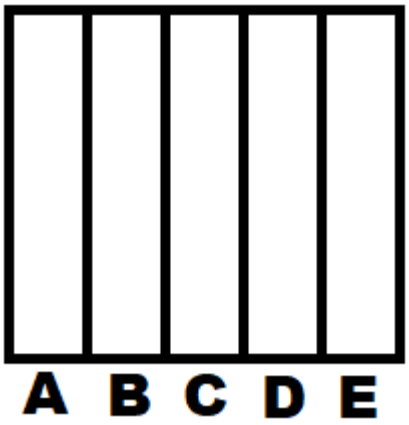
Se o leitor de códigos que utilize apenas cinco barras, a quantidade de códigos com leitura da esquerda para a direita igual à da direita para a esquerda, desconsiderando todas as barras claras ou todas as escuras, é:

a) 14
b) 12

c) 8
d) 6
e) 4

Resolução:

Chamemos as cinco barras do código pelas letras maiúsculas A, B, C, D e E como é mostrado na figura a seguir.



Como cada barra pode ser preenchida com a cor preta ou não, sendo assim, temos 2 opções de preenchimento para cara uma. Porém, para que a leitura seja feita igualmente tanto da esquerda para a direita quanto da direita para a esquerda teremos algumas restrições

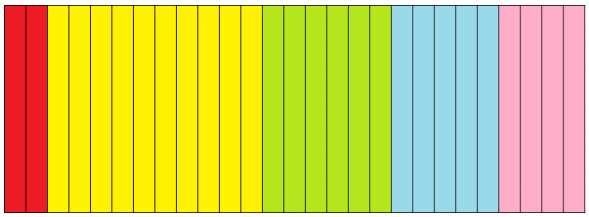
- A e E precisam ter a mesma cor, 2 opções;
- B e D precisam ter a mesma cor, 2 opções;
- C pode ter qualquer cor, 2 opções;

Sendo assim, teríamos $2 \times 2 \times 2 = 8$ códigos de leitura que seguem o padrão desejado, mas 2 dentre os 8 serão os que possuem todas as cores iguais, ou seja, todos brancos ou todos pretos.

Logo, a resposta será $8 - 2 = 6$.

Atividade 5

Cada aluno deverá codificar o código de barras a seguir com as informações



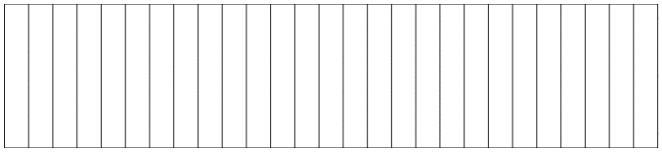
- Barras 1 e 2: turno: 1º, 2º e 3º. (Barras vermelhas)
  Deve ter dois algarismos.
- Barras 3 a 12: número da turma (611, 612, 613, 614, 615, 621, 622, 623, 624, 625). (Barras amarelas)
  Deve ter dez algarismos.
- Barras 13 a 18: número da chamada: 1 a 40. (Barras verdes)
  Deve ter seis algarismos.
- Barras 19 a 23: Dia do aniversário do aluno. (Barras azuis)
  Deve ter cinco algarismos
- Barras 24 a 27: Mês do aniversário do aluno. (Barras rosas)
  Deve ter quatro algarismos

* Se o número decimal após a conversão para binário não possuir a quantidade de dígitos necessária segundo os requisitos acima, deve-se adicionar zeros à esquerda até atingir o valor necessário.

Exemplo: Os números da chamada vão de 1 a 40 e devem ter seis algarismos. Um aluno que possui o número 15 deve convertê-lo para binário e encontrar $(1111)_2$, porém, como esse valor só tem 4 algarismos, então é necessário adicionar 2 zeros à esquerda para ficar no padrão da atividade, $(001111)_2$.

As barras de valor 1 deverão ser pintadas de preta e as barras de valor zero deverão permanecer brancas. Agora, preencha a tabela abaixo com seus dados, converta-os para binário e, após isso, faça a pintura do código de barras.

	Decimal	Binário
Turno		
Turma		
Nº da chamada		
Dia do Aniversário		
Mês do aniversário		



Resolução:

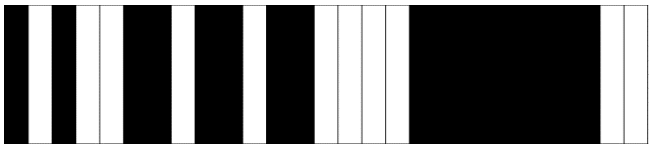
O aluno Alberto estuda na turma 621 (2º turno) e possui o 33 como número da chamada, e também seu aniversário é na data de 31/12. Sua tabela será preenchida da seguinte forma.

	Decimal	Binário
Turno	2	10
Turma	621	1001101101
Nº da chamada	33	100001
Dia do Aniversário	31	11111
Mês do aniversário	12	1100

Colocando os valores binários na ordem de preenchimento do código de barras, temos:

10 1001101101 100001 11111 1100

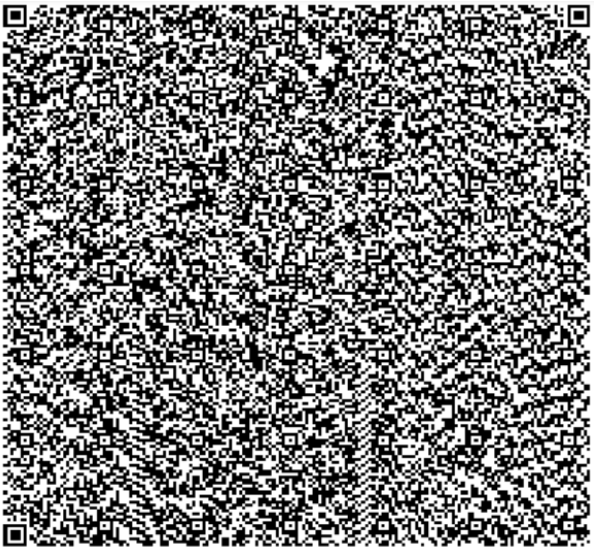
Agora, pintamos de preto as barras relacionadas com os algarismos 1 da sequência.



A imagem acima é o resultado da conversão dos dados do aluno em um código de barras com as características da atividade.

Atividade 6

Um QR Code Model 2 é uma figura quadrada, pois cada linha e cada coluna possui a mesma quantidade de pixels (quadradinhos). Este modelo de QR Code é dividido em quarenta versões que vão de 1 a 40. Na Versão 1 temos 21 por 21 pixels totalizando 441 quadradinhos, na Versão 2 temos 25 por 25 e, para cada versão maior, a quantidade de pixels, por linha e coluna, aumenta sempre 4 unidades.



Acima, vemos a Versão 1 (21x21 pixels), à esquerda, e a Versão 40 (177x177 pixels), à direita. Com base no crescimento da quantidade de pixels em um QR Code, podemos afirmar que é:

a) Linear
b) Exponencial
c) Quadrático
d) Logarítmico
e) N.R.A.

Resolução:

Como foi dito no texto, partindo da Versão 1 (21x21), cada versão do QR Code aumenta de quatro em quatro a quantidade de pixels nas linhas e colunas.

$$Versão\ 1 - 21 \times 21 - 441\ pixels$$

$$Versão\ 2 - 25 \times 25 - 625\ pixels$$

$$Versão\ 3 - 29 \times 29 - 814\ pixels$$

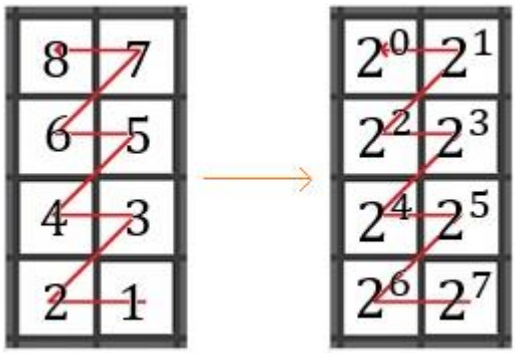
Até

$$Versão\ 40 - 177 \times 177 - 31329\ pixels$$

Logo, a quantidade de pixels em uma imagem de QR Code é sempre um quadrado perfeito e, com isso, seu crescimento ao longo das versões é quadrático.

Atividade 7

Em um QR Code, um byte pode ser um bloco de 8 bits (quadradinhos) que contém um caractere como mensagem cada. Cada bit pode receber a cor branca ou preta que correspondem, respectivamente, a zero ou um. Dentro de cada byte a pintura das cores correspondem a seguinte ordem.



Ao lado, vemos que o primeiro bit pintado corresponde a potência 2^7, o segundo a 2^6, e assim por diante até chegarmos a potência 2^0 que corresponde ao oitavo bit.

Vamos utilizar a tabela de codificação abaixo para consulta dos valores associados a cada caractere que utilizaremos.

Decimal	Caractere	Binário:	Decimal	Caractere	Binário:
48	0	00110000	105	i	01101001
49	1	00110001	106	j	01101010
50	2	00110010	107	k	01101011
51	3	00110011	108	l	01101100
52	4	00110100	109	m	01101101
53	5	00110101	110	n	01101110
54	6	00110110	111	o	01101111
55	7	00110111	112	p	01110000
56	8	00111000	113	q	01110001
57	9	00111001	114	r	01110010
97	a	01100001	115	s	01110011
98	b	01100010	116	t	01110100
99	c	01100011	117	u	01110101
100	d	01100100	118	v	01110110
101	e	01100101	119	w	01110111
102	f	01100110	120	x	01111000
103	g	01100111	121	y	01111001
104	h	01101000	122	z	01111010

Note que, a letra “a” está associada ao valor 97 que em binário corresponde a $(1100001)_2$, como este valor possui sete algarismos, adicionamos um caractere de valor zero à esquerda dele formando assim $(01100001)_2$. Pois, cada byte precisa ter “tamanho” de 8 bits.

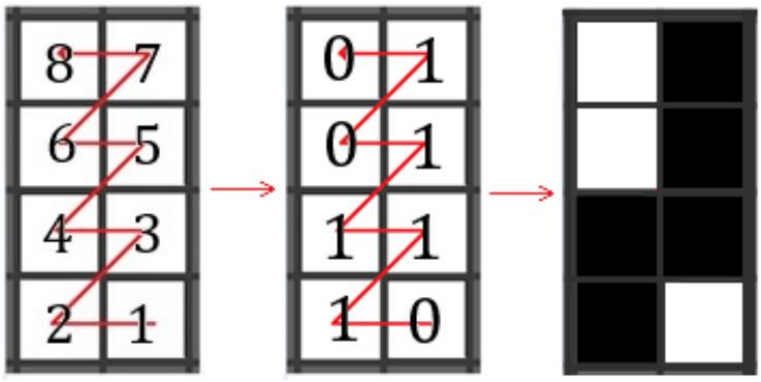
Agora, veja um exemplo de preenchimento de um byte.

Um byte terá como mensagem a letra “z” que na codificação do Modo Byte terá o valor de 01111010, se sua escrita é como foi mostrado anteriormente, como ficará o preenchimento?

Primeiro, para facilitar, organizamos os bits de 1 a 8, como a tabela a seguir.

1	2	3	4	5	6	7	8
0	1	1	1	1	0	1	0

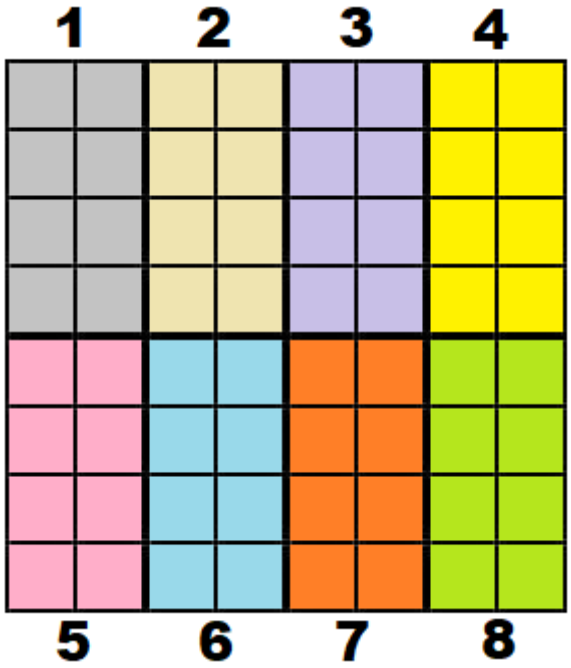
Vamos agora preencher cada quadradinho da imagem a seguir de acordo com a numeração da primeira linha da tabela anterior. Analisando a segunda linha, cada bit que estiver relacionado com o dígito 1 será pintado de preto.



Agora, iremos realizar a pintura de uma figura de 8 bytes com a mensagem “educador”. Cada letra da palavra a ser codificada será associada ao byte cujo valor corresponde a sua posição dentro da palavra.

e (byte1), d (byte 2), u (byte 3), c (byte 4)

a (byte 5), d (byte 6), o (byte 7), o (byte 8)



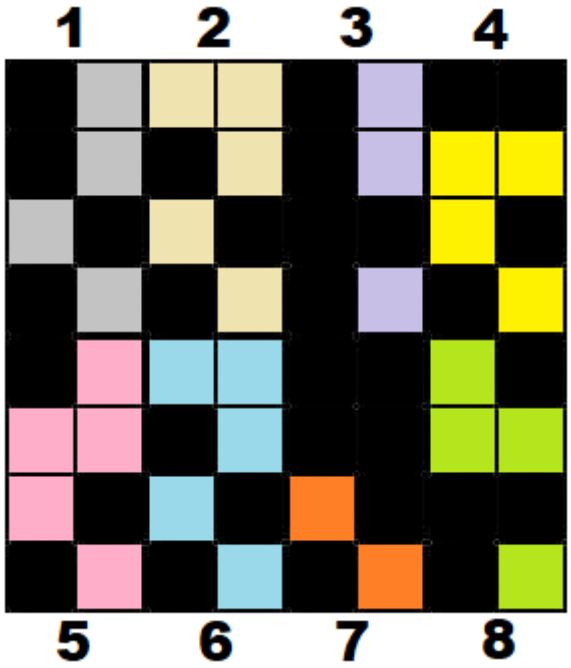
Os bits estão coloridos para facilitar a identificação de cada byte. Na resolução, vamos pintar apenas os bits de cor preta.

Resolução:

Primeiro verificamos o valor que cada letra está associada nos números decimais. Após isso fazemos a conversão de cada valor para binário. Veja a tabela a seguir.

e	101	01100101	a	97	01100001
d	100	01100100	d	100	01100100
u	117	01110101	o	111	01101111
c	99	01100011	r	114	01110010

Por último, cada letra será pintada seguindo a ordem posicional.



Atividade 8

Utilizando a câmera do celular ou um aplicativo adequado cada aluno deve escanear o QR Code abaixo e resolver a expressão dada.

Ordem das operações:

1º) Potenciação e Radiciação

2º) Multiplicação e Divisão

3º) Soma e Subtração

Ordem dos símbolos:

1º) as operações que estão dentro dos parênteses

2º) as operações que estão dentro dos colchetes

3º) as operações que estão dentro das chaves

Resolução:

Após escanear a imagem o aluno encontrará a seguinte expressão:

$$\{[(2 + 5 \times 3) \times 2 - 7] \times 10 + 1\} + 16$$

Resolvendo dentro dos parênteses, temos:

$$\{[17 \times 2 - 7] \times 10 + 1\} + 16$$

Resolvendo dentro dos colchetes, temos:

$$\{27 \times 10 + 1\} + 16$$

Resolvendo dentro das chaves, temos:

271 + 16

E finalmente

287

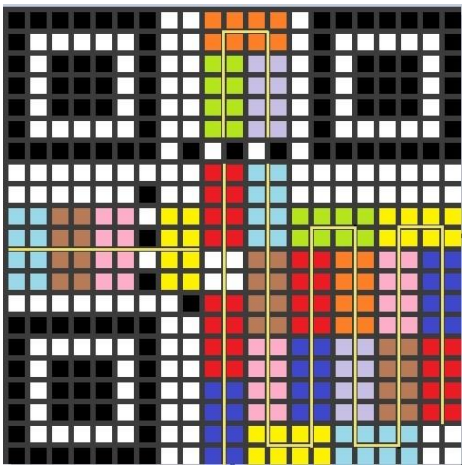
Agora, cada aluno deve ir apresentando suas soluções ao professor e ele, por sua vez, deve mostrar um desses QR Codes abaixo para o aluno dependendo de sua resposta.


Correto


Errado

Os códigos não deverão ter uma legenda como a mostrada acima.

Atividade 9

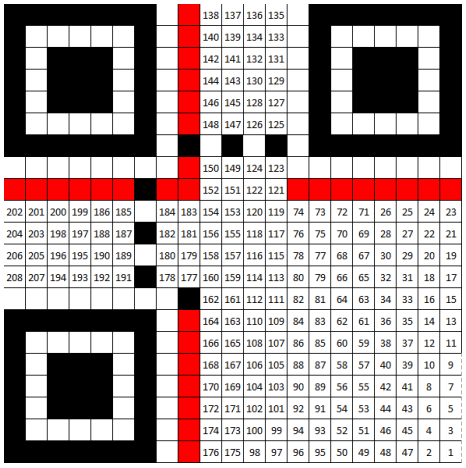


O QR Code da versão 1 possui $21 \times 21 = 441$ pixels (bits), que são quadradinhos pretos ou brancos, sendo que o quadrado subdividido em quatro bits na parte inferior direita indica o conteúdo desta atividade que será letras e números no Modo Byte, nos bits 1, 2, 3 e 4, e serão marcados com os dígitos 0100, e a parte colorida indica onde a mensagem será alocada na imagem. Cada parte da mensagem é subdividida em 8 quadradinhos (byte) que serão preenchidos de acordo com uma ordem específica (segunda imagem). Os bits brancos e pretos indicam os dígitos binários 0 e 1, respectivamente. Os bits de 5 a 12 (primeiro byte) indicam quantos caracteres estarão presentes na mensagem, por exemplo, se contém a palavra “escola ” então o primeiro byte terá o número 7 como mensagem, que corresponde ao primeiro byte mais seis bytes da “escola”. (Se o valor for maior

Decimal	Caractere	Binário:	Decimal	Caractere	Binário:
48	0	00110000	105	i	01101001
49	1	00110001	106	j	01101010
50	2	00110010	107	k	01101011
51	3	00110011	108	l	01101100
52	4	00110100	109	m	01101101
53	5	00110101	110	n	01101110
54	6	00110110	111	o	01101111
55	7	00110111	112	p	01110000
56	8	00111000	113	q	01110001
57	9	00111001	114	r	01110010
97	a	01100001	115	s	01110011
98	b	01100010	116	t	01110100
99	c	01100011	117	u	01110101
100	d	01100100	118	v	01110110
101	e	01100101	119	w	01110111
102	f	01100110	120	x	01111000
103	g	01100111	121	y	01111001
104	h	01101000	122	z	01111010

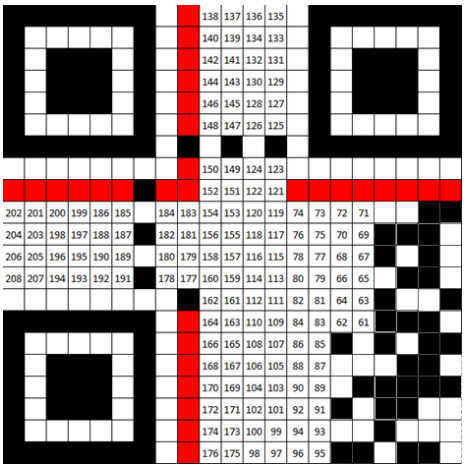
do que 9 então basta transformar o decimal para binário e preenche-lo à esquerda para ficar com 8 algarismos de tamanho).

A partir do bit 13 em diante até o bit 208 será escrita a mensagem desejada. Por exemplo, codificando um QR Code com a mensagem “escola” no Modo Byte e com base na tabela ao lado, teremos a seguinte ordem dos bits.



Modo Byte	1º Byte (7)	e	s	c	o	l	a
0100	00110111	01100101	01110011	01100011	01101111	01101100	01100001

Desta forma, utilizaremos os 60 primeiros bits do QR Code e pintaremos de preto os bits relacionados aos dígitos 1. A tabela abaixo está preenchida com os 60 algarismos da tabela anterior relacionando-os à numeração do bit correspondente.



1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
0	1	0	0	0	0	1	1	0	1	1	1	0	1	1	0	0	1	0	1	0	1	1	1	0	0	1	1	0	1
31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60
1	0	0	1	1	1	0	1	1	0	1	1	1	1	0	1	1	0	1	1	0	0	0	1	1	0	0	0	0	1

O resultado será o QR Code pintado ao lado.

Com base nas informações apresentadas, preencha o QR Code a seguir com a mensagem “mestradoprofmatunirio” no Modo Byte (tabela de codificação da página anterior). Utilize as tabelas a seguir como guia para o preenchimento.

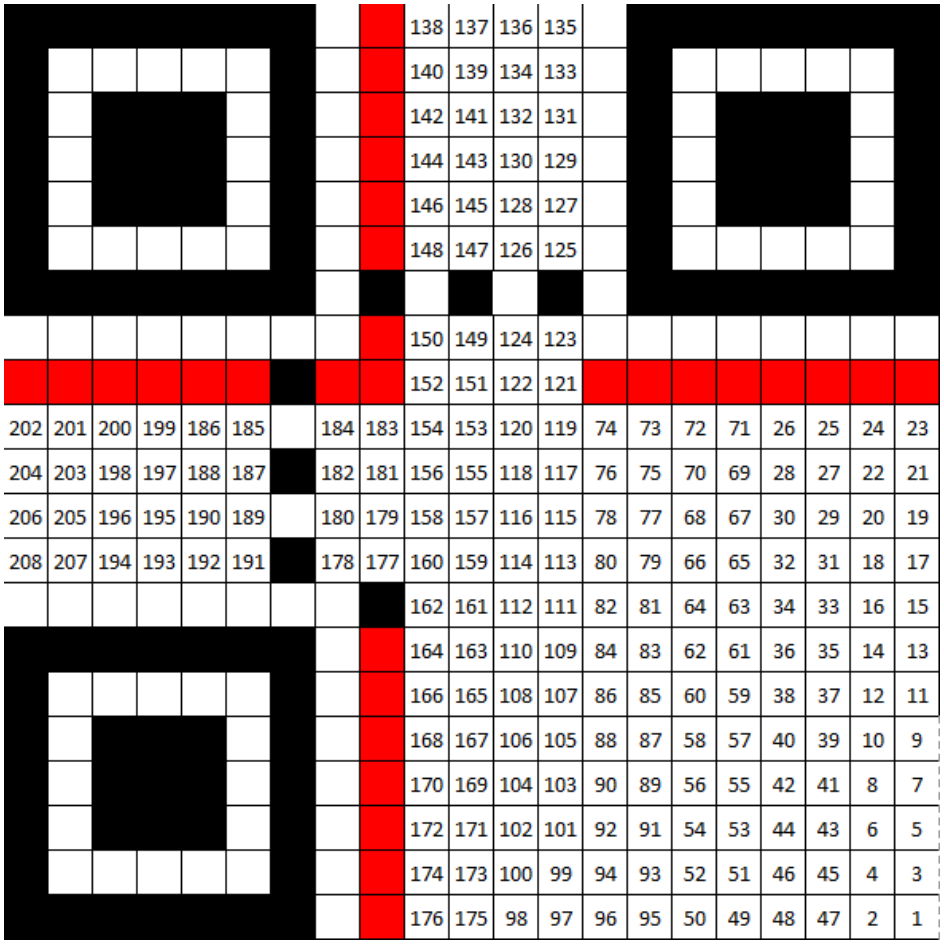
Preencha a tabela abaixo com os valores binários do Modo Byte, do primeiro byte e dos caracteres.

Modo Byte	1º Byte	m	e	s	t	r	a	d	o	p	r
o	f	m	a	t	u	n	i	r	i	o	

Complete a tabela abaixo com os valores de cada bit segundo a tabela acima.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90
91	92	93	94	95	96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111	112	113	114	115	116	117	118	119	120
121	122	123	124	125	126	127	128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143	144	145	146	147	148	149	150
151	152	153	154	155	156	157	158	159	160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175	176	177	178	179	180

Pinte de preto os quadradinhos do QR Code correspondentes ao algarismo 1.



Resolução: Vamos preencher a primeira tabela com os valores dos caracteres no modo binário.

1º) Preencher o valor do Modo Byte (0100)

2º) Preencher o primeiro byte com o valor correspondente ao número 23 em binário (10111), sendo que, como o o número precisar ter tamanho de 8 bits, então adicionamos 3 zeros à esquerda, e finalmente temos (00010111).

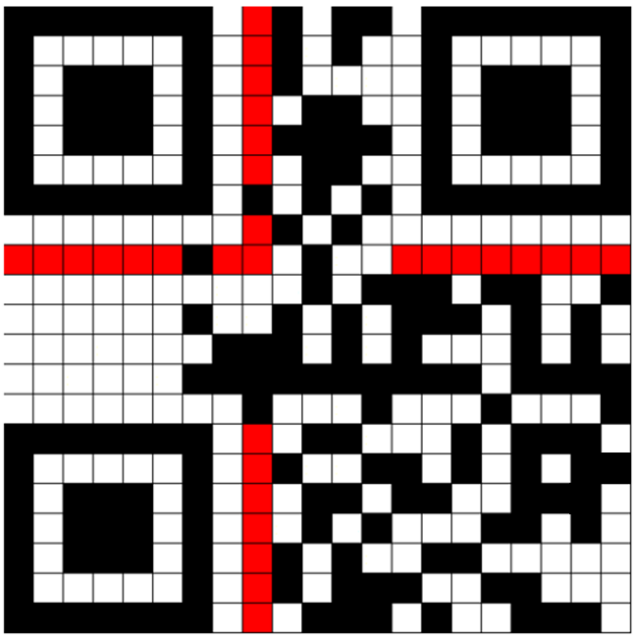
3º) Preencher o restante da tabela com o valor correspondente as letras.

Modo Byte	1º Byte	m	e	s	t	r	a	d	o	p	r
0100	00010111	01101101	01100101	01110011	01110100	01110010	01100001	01100100	01101111	01110000	01110010
o	f	m	a	t	u	n	i	r	i	o	
01101111	01100110	01101101	01100001	01110100	01110101	01101110	01101001	01110010	01101001	01101111	

O passo seguinte é preencher a segunda tabela com os 180 bits da atividade.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
0	1	0	0	0	0	0	1	0	1	1	1	0	1	1	0	1	1	0	1	0	1	1	0	0	1	0	1	0	1
31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60
1	1	0	0	1	1	0	1	1	1	0	1	0	0	0	1	1	1	0	0	1	0	0	1	1	0	0	0	0	1
61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90
0	1	1	0	0	1	0	0	0	1	1	0	1	1	1	1	0	1	1	1	0	0	0	0	0	1	1	1	0	0
91	92	93	94	95	96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111	112	113	114	115	116	117	118	119	120
1	0	0	1	1	0	1	1	1	1	0	1	1	0	0	1	1	0	0	1	1	0	1	1	0	1	0	1	1	0
121	122	123	124	125	126	127	128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143	144	145	146	147	148	149	150
0	0	0	1	0	1	1	1	0	1	0	0	0	1	1	1	0	1	0	1	0	1	1	0	1	1	1	0	0	1
151	152	153	154	155	156	157	158	159	160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175	176	177	178	179	180
1	0	1	0	0	1	0	1	1	1	0	0	1	0	0	1	1	0	1	0	0	1	0	1	1	0	1	1	1	1

Agora, vamos pintar de preto cada quadradinho correspondente ao algarismo 1 na tabela acima.



Este será o resultado final da pintura, os bits brancos foram pintados apenas para ilustração.

Conclusão

Esperamos que esse trabalho possa contribuir para os docentes que se interessem por novas tecnologias nas salas de aula do Ensino Básico e que possa ajudar os alunos a compreenderem melhor os códigos que nos cercam diariamente. Segundo Veiga (1996), “O trabalho docente alienado só pode gerar um produto discente alienado; se isso não acontece é porque o aluno conseguiu, por outros caminhos, criticar a prática de seu professor. É por esse motivo que afirmamos que o professor precisa saber como se constitui o conhecimento. Caso contrário, poderá não só tornar inócuo o processo de aprendizagem, como até obstruir o processo de desenvolvimento que o fundamenta. ”(VEIGA, 1996: p. 68-69).

Está cada vez claro para nós que estamos inseridos em uma sociedade muito tecnológica e que a educação precisa acompanhar essas evoluções, trazendo para dentro da sala de aula algumas inovações, com a finalidade de formar jovens capazes de compreender o mundo que o cerca e ter um diferencial no mercado de trabalho cada vez mais disputado. Além disso, uma aula com códigos poderá despertar no aluno a curiosidade de aprender mais sobre o tema e, assim, o discente terá a possibilidade de trilhar sua carreira no mundo matemático e/ou informático.

Também esperamos que este trabalho sirva de material bibliográfico para o professor se aprofundar no assunto.

Apêndice
Sistema Binário de Numeração

O sistema de numeração de base 2, ou sistema binário, é um sistema de numeração posicional em que todos os números são representados por apenas dois algarismos, zero e um. Por exemplo, $100_{(2)}, (100)_2$, onde o algarismo 2 representa a base e lê-se um, zero, zero na base 2.

Apesar de existir há muito tempo e ter sido utilizado por algumas civilizações, esse sistema ganhou mais notoriedade após o advento da computação, pois os dispositivos utilizam esse sistema para realização de cálculos e processamento de dados em alta velocidade com *hardwares* mais simples.

Este sistema de numeração, assim como os outros, também possui as definições de ordem e classe que veremos a seguir.

- **Ordem:** é a posição ocupada por cada símbolo em um número.
- **Classe:** é a reunião de três ordens, sendo que apenas a classe na extrema esquerda pode ter duas ou uma ordem.

Para representarmos um número genericamente na base dois devemos seguir o mesmo modo da base decimal, ou seja, um número com $n+1$ algarismos será mostrado da seguinte forma

$$a_n \cdot 2^n + a_{n-1} \cdot 2^{n-1} + \cdots + a_2 \cdot 2^2 + a_1 \cdot 2^1 + a_0 \cdot 2^0 = (a_n a_{n-1} \dots a_1 a_0)_2$$
$$= a_n a_{n-1} \dots a_1 a_{0_{(2)}}$$

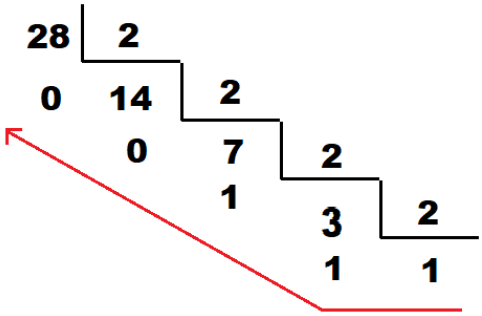
Esta é a expansão de um número natural na base 2 em que cada dígito $a_i, 0 \leq i \leq n$ é igual a 0 ou 1.

E esta forma de representar ajuda a transformar um número entre a base 2 e a base 10.

Exemplo 1:

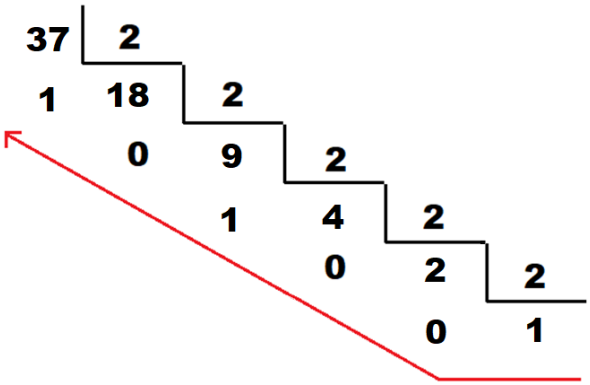
- $11100_{(2)} = 1 \cdot 2^4 + 1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 0 \cdot 2^0 = 16 + 8 + 4 = 28$
- $100101_{(2)} = 1 \cdot 2^5 + 0 \cdot 2^4 + 0 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 32 + 4 + 1 = 37$

Uma maneira de transformar um número da base 10 para a base binária é a das divisões euclidianas sucessivas, na qual devemos dividir o número desejado por 2 e, em seguida, ir dividindo os quocientes das divisões também por dois até encontrarmos um quociente menor ou igual a 1. Veja a seguir



Sendo assim o número será formado pelo último quociente e os restos das divisões anteriores na ordem mostrada acima. Logo temos que $28 = 11100_{(2)}$.

Da mesma forma, convertendo o número 37 em um número binário, temos



Logo, $37 = (100101)_2$.

Referências Bibliográficas

Pinto, Ana C. M.; Felcher, Carla D. O.; Ferreira, André L. A. CONSIDERAÇÕES SOBRE O USO DO APLICATIVO QR CODE NO ENSINO DA MATEMÁTICA: REFLEXÕES SOBRE O PAPEL DO PROFESSOR, São Paulo, 2016. Disponível em <http://www.sbem.com.br/enem2016/anais/pdf/8323_4386_ID.pdf>. Acesso 20 jan. 2021.

SADOVSKY, Patrícia. Falta fundamentação didática no ensino de matemática. Revista Nova Escola, Editora Abril, São Paulo. Ed.Especial14.p.08-10.Jul.2007.

GS1BR - <Conhecendo o código de barras – Vol. 1>. Disponível em <https://www.gs1br.org/codigos-e-padroes/padroes-de-identificacao/MateriaisGTIN/Conhecendo%20o%20C%C3%B3digo%20de%20Barras.pdf> Acesso em 24/05/2021.

Silva, Thayany B. da; Bezerra, Simone M. C. B. O USO DO QR CODE NO ENSINO DE MATEMÁTICA NA FORMAÇÃO INICIAL. X Simpósio Linguagens e Identidades da/na Amazônia sul-ocidental VIII Colóquio Internacional "As Amazônias, as Áfricas na Pan-Amazônia", 2016.

POLCINO MILIES, C. A Matemática dos códigos de barras, Revista do Professor de Matemática (RPM) 65.

POLCINO MILIES, C. A Matemática dos códigos de barras, Revista do Professor de Matemática (RPM) 68.

GS1BR - <https://www.gs1br.org/codigos-e-padroes/captura/Paginas/GS1-QR-Code.aspx>.

Referências das imagens

[1] Disponível em <https://www.i9automacaocomercial.com.br/blog/descubra-quem-inventou-o-codigo-de-barras/>. Acesso 20 jan. 2021.

[2] Disponível em <https://www.gs1br.org/codigos-e-padroes/captura#codigos-de-barra>. Acesso 20 jan. 2021.

[3] Disponível em <http://wiki.biserp.com.br/index.php/Especifica%C3%A7%C3%A3o_de_Guias_e_Boletos>. Acesso em 21 jan. 2021.

[4] Disponível em <https://www.shopify.com.br/blog/69456645-guia-completo-sobre-codigo-de-barras-no-brasil>. Acesso em 21 jan. 2021.

[5] Disponível em <https://rpm.org.br/cdrpm/65/9.html>. Acesso 22 jan. 2021.

[6] Disponível em <https://www.thonky.com/qr-code-tutorial/module-placement-matrix>. Acesso em 23 jan. 2021.

[7] Disponível em <https://pt.wikipedia.org/wiki/C%C3%B3digo_QR>. Acesso em 23 jan. 2021.

[8] Disponível em <https://commons.wikimedia.org/wiki/File:QR_Code_Mask_Patterns.svg>. Acesso em 31 jan. 2021.

[9] Disponível em <https://www.youtube.com/watch?v=142TGhaTMtI>. Acesso em 25 jan. 2021.

[10] Disponível em <https://www.youtube.com/watch?v=oR9D95XOnkc>. Acesso em 25 jan. 2021.

[11] Disponível em <https://www.thonky.com/qr-code-tutorial/format-version-information>. Acesso em 20 fev. 2021.

[12] Disponível em <https://www.thonky.com/qr-code-tutorial/data-analysis>. Acesso em 20 fev. 2021.