

Transmissão de Dados do Eleitor via Múltiplos QR Codes para Mobilidade em Votações

Carolina Borzino Rocha¹, Leonardo Kimura¹, Marcos Simplicio Jr¹

¹Escola Politécnica – Universidade de São Paulo (USP)

{crocha, lkimura, mjunior}@larc.usp.br

Abstract. *Anywhere voting solutions enable voters to cast ballots at any polling station, providing a promising alternative for increasing participation in electronic voting elections. However, current proposals require the transmission of voters' biometric data to voting machines through voting tokens, such as QR codes. Since transmitting large volumes of data in this manner is nontrivial, this work investigates the feasibility of transmitting biometric data using multiple QR codes displayed sequentially in video format, considering an offline scenario and the use of low-cost cameras. Different QR code densities and display rates were evaluated. The results indicate that approximately 12 KB of data can be transmitted in ≈ 5 seconds.*

Resumo. *Existem soluções de mobilidade que permitem aos eleitores votar em qualquer seção eleitoral, criando uma alternativa promissora para aumentar a participação em eleições com votação eletrônica. Entretanto, propostas atuais exigem a transmissão de dados biométricos dos eleitores para as urnas por meio de tokens, como QR codes. Como a transmissão de grandes volumes de dados dessa maneira não é trivial, este trabalho analisa a viabilidade da transmissão de dados biométricos utilizando múltiplos códigos exibidos em vídeo, em um cenário sem conexão à Internet e com câmeras de baixo custo. Avaliando diferentes densidades de QR codes e taxas de exibição, os resultados indicam que é possível transmitir aproximadamente 12 KB de dados em ≈ 5 segundos.*

1. Introdução

Muitas democracias no mundo sofrem com elevadas taxas de abstenção de votantes em seus processos eleitorais. No Brasil, por exemplo, essa taxa chegou a 21,68% do eleitorado em 2024 (32 milhões) [Tribunal Superior Eleitoral 2025]. Embora existam muitos motivos para a abstenção, incluindo potencial desinteresse, um dos mais relevantes é a distância do eleitor até a seção eleitoral na qual está registrado [Folha de S.Paulo 2022]. Nas eleições da Grécia em 2023, por exemplo, esta foi a razão de 31% das abstenções envolvendo motivos práticos [Heinrich Böll Stiftung Greece 2024]. É possível dizer, assim, que uma maneira potencialmente eficaz de aumentar a participação eleitoral consiste em prover maior mobilidade na votação, permitindo que eleitores votem em qualquer seção eleitoral.

Diversas soluções de mobilidade na votação exigem que eleitores móveis se autenticem junto aos equipamentos por meio de “tokens de votação”, os quais carregam seus modelos (*templates*) biométricos assinados digitalmente. Esta estratégia é especialmente comum em cenários nos quais as urnas eletrônicas: (1) não possuem conexão à Internet,

como é o caso dos equipamentos usados na Índia e no Brasil; e (2) não carregam dados de autenticação de todos os eleitores aptos a votar, em particular devido à grande escala do corpo eleitoral. De fato, esse tipo de token aparece em algumas soluções de mobilidade listadas em [Kimura and Simplicio Jr. 2025], que tem como foco o cenário brasileiro.

Embora esta abordagem baseada em tokens possa parecer simples à primeira vista, na prática não é trivial transmitir uma grande quantidade de dados para urnas eletrônicas de maneira segura, transparente e com elevada usabilidade. Mecanismos como Bluetooth, Wi-Fi e NFC (*Near Field Communication*) comumente são evitados para preservar o caráter “offline” das urnas eletrônicas, importante para limitar a superfície de ataque a esses equipamentos. Ao mesmo tempo, transmissões baseadas em Bluetooth e Wi-Fi costumam ser mais suscetíveis a ataques de interceptação e retransmissão, ou negação de serviço, enquanto o suporte a NFC não é tão ubíquo em dispositivos celulares de baixo custo. Neste cenário, o uso de códigos de barras bidimensionais, também conhecidos como *quick-response (QR) codes*, apresenta-se como uma alternativa interessante ao combinar baixo acoplamento físico com ampla disponibilidade [Kimura and Simplicio Jr. 2025].

Ainda assim, o uso de QR codes exibidos visualmente também apresenta uma superfície de ataque própria, sujeita a riscos como a captura não autorizada da tela durante a exibição do código ou a substituição do QR code legítimo por um falso. Neste trabalho, assume-se que os dados transmitidos já chegam cifrados e assinados digitalmente ao processo de geração do QR code, de forma que a mitigação desses riscos depende primariamente de controles no dispositivo emissor, não do canal de transmissão em si.

Tendo em vista o interesse prático em usar QR codes no cenário de mobilidade da votação, este trabalho investiga a viabilidade de transmitir grandes quantidades de dados através dessa tecnologia, como um estudo inicial que considera os obstáculos práticos e de segurança envolvidos nesse cenário. Análises mais aprofundadas sobre vetores de ataque específicos e a validação em outros dispositivos de captura são deixadas para investigações futuras.

Como um único QR code armazena tipicamente uma quantidade limitada de dados, insuficiente para comportar modelos biométricos em um único código, considera-se a estratégia de dividir os dados a serem transmitidos em múltiplos QR codes, apresentados e lidos no formato de vídeo. Foram testadas diferentes densidades de QR codes e velocidades de exibição, usando uma câmera web de baixo custo para a sua leitura e decodificação. Os testes realizados indicam que é possível transmitir 12 KB de dados em apenas ≈ 5 segundos, o que pode ser considerado viável para um contexto de uso de mobilidade de votação em larga escala.

2. Conceitos teóricos

Esta seção apresenta os principais conceitos relacionados à mobilidade na votação e discute trabalhos correlatos envolvendo transmissão de dados e autenticação por meio de QR codes.

2.1. Mobilidade na votação

Mobilidade na votação (em inglês, *anywhere voting*) permite o voto em qualquer seção eleitoral, sendo particularmente benéfica para eleitores que estão viajando ou que se mudaram recentemente. Países que implementaram mobilidade na votação,

como Estados Unidos e Canadá, observaram reduções significativas nas abstenções [Chief Election Officer 2015, Stein and Vonnahme 2012]. Entretanto, tipicamente as implementações atuais de mobilidade na votação exigem autenticar os eleitores em aparelhos conectados à Internet para evitar voto duplo. Este requisito torna-se problemático quando questões de infraestrutura e/ou segurança limitam (ou, no caso brasileiro, impedem) o acesso à rede pelas urnas eletrônicas.

O trabalho [Kimura and Simplicio Jr. 2025] apresenta algumas alternativas para implementar mobilidade na votação no contexto brasileiro, discutindo seus benefícios e limitações. Um exemplo lá discutido envolve o uso de *mixnets* e decifração de limiar (*threshold decryption*). Nele, os eleitores se autenticam na urna usando tokens de votação contendo não só templates biométricos, mas tokenID únicos por eleitor. As urnas verificam o token, autenticam biometricamente o eleitor e recebem o voto, que é cifrado e associado ao tokenID. Todos os votos cifrados com seus respectivos tokenID são enviados para o órgão de administração eleitoral, que detecta e remove os votos com o mesmo tokenID. Os votos cifrados restantes são então embaralhados através de uma *mixnet* e decifrados usando um mecanismo de *threshold decryption*.

2.2. Trabalhos relacionados

Historicamente, a utilização de QR codes na literatura evoluiu de aplicações simples de rastreamento e identificação para cenários mais complexos envolvendo autenticação, transmissão de arquivos e troca rápida de informações [Tiwari 2016]. Entretanto, a transmissão de grandes volumes de dados utilizando QR codes ainda apresenta desafios relacionados principalmente à capacidade limitada de armazenamento por código, à confiabilidade da captura e às restrições impostas pelos dispositivos de leitura.

Nesse contexto, [Rizky and Munir 2023] propuseram um sistema de transmissão de dados utilizando múltiplos QR codes exibidos sequencialmente. O trabalho tem como principal objetivo maximizar a taxa de transmissão de dados, utilizando codificação RaptorQ para tolerância à perda de quadros e permitindo a reconstrução da informação mesmo quando alguns QR codes não são capturados corretamente. Apesar de eficiente para transmissão de arquivos em dispositivos móveis, o estudo assume condições relativamente favoráveis de captura e não discute limitações impostas por câmeras de baixo custo ou por ambientes com restrições operacionais mais severas.

De maneira semelhante, [Heng et al. 2021] investigam a transmissão de arquivos via QR codes em ambientes complexos, analisando principalmente os impactos de iluminação, movimento e qualidade da captura na taxa de sucesso da decodificação. Os autores propõem estratégias que aumentam a robustez da leitura, ainda que em detrimento do tempo total de transmissão. O estudo destaca que a qualidade da câmera influencia diretamente na confiabilidade da leitura, indicando que QR codes muito densos tornam-se difíceis de capturar em dispositivos com resolução limitada.

Já no contexto de autenticação segura, [Mahansaria and Roy 2020] exploraram o uso de QR codes contextuais de uso único para autenticação baseada em senhas de uso único. Nesse sistema, QR codes são utilizados como mecanismo de segurança para troca de informações criptográficas e autenticação do usuário. Entretanto, o volume de dados transmitido é relativamente pequeno quando comparado ao necessário para transmissão de templates biométricos completos.

Diferentemente dos trabalhos anteriores, o presente estudo não busca apenas maximizar a taxa de transmissão ou propor mecanismos de autenticação baseados em QR codes. O foco principal é avaliar a viabilidade prática da transmissão rápida de aproximadamente 12 kB de dados biométricos cifrados em um cenário offline e em larga escala. Ao mesmo tempo, investigam-se especificamente os limites de desempenho utilizando câmeras de baixo custo e resolução limitada, condição mais compatível com a realidade operacional de sistemas eleitorais em larga escala. Assim, a principal contribuição deste trabalho consiste em analisar o equilíbrio entre densidade dos QR codes, tempo de transmissão e confiabilidade de leitura em condições compatíveis com a mobilidade na votação.

3. Requisitos

A fim de estabelecer os requisitos para este estudo, analisaram-se as condições necessárias à implementação no caso brasileiro, já que o país conduz a maior eleição com votação eletrônica do mundo [Tribunal Superior Eleitoral 2023].

Para as eleições do Brasil, o token a ser transmitido às urnas eletrônicas deve conter integralmente as informações necessárias para a validação de integridade e autenticidade das informações do eleitor e de seu *template* biométrico. Dentre essas informações, destacam-se o identificador único do token, a zona e seção a que corresponde, a assinatura digital e o *template* biométrico do eleitor, cifrado previamente pelo próprio TSE, de modo que apenas as urnas legítimas — detentoras da chave correspondente — sejam capazes de decifrá-lo e validá-lo.

Para estimativa do volume total de dados a serem transferidos, foi utilizado como referência o *template* biométrico, que corresponde ao conjunto de informações de maior tamanho e, conseqüentemente, ao caso com transmissão mais complexa. Considerando que *templates* modernos tipicamente contêm alguns kilobytes por dedo [Engelbrecht et al. 2023], adotou-se para testes o padrão de aproximadamente 3 kB de dados por dedo cuja biometria foi cadastrada. Como o sistema empregado pelo Tribunal Superior Eleitoral utiliza o cadastro de quatro dedos (impressões digitais dos polegares e indicadores) [TRE-SP 2024], o volume total de dados biométricos transmitidos para a urna é próximo de 12 kB. Neste estudo, foi analisada a viabilidade da transmissão de 12 kB de dados a partir dos QR codes.

A viabilidade da proposta de mobilidade na votação com autenticação a partir de QR codes depende diretamente do tempo necessário para a leitura completa dos dados. Como a proposta é de que cada eleitor que optar pelo voto móvel deverá apresentar seu token no terminal do mesário, a decodificação não pode ser longa. Caso contrário, o processo de votação torna-se mais demorado, prejudicando o avanço ideal da eleição.

A codificação do volume de 12 kB de dados resulta em alguns QR codes muito densos e, portanto, de leitura mais difícil por dispositivos. Aumentando-se o número de códigos em que as informações estão divididas, a densidade de cada um é reduzida. Assim, faz-se necessário mostrar os diferentes QR codes gerados para a leitura do conjunto completo de dados. Porém, caso a passagem dos diferentes QR codes seja feita de forma manual, o tempo necessário para autenticação torna-se mais longo e o procedimento, mais suscetível a falhas humanas. Portanto, este trabalho propõe o sequenciamento dos QR codes em vídeo a fim de automatizar e acelerar o processo.

Assim, o tempo necessário para leitura completa dos QR codes torna-se um parâmetro crítico para a viabilidade da proposta. Como o processo de autenticação já envolve a validação biométrica do eleitor, tempos excessivamente longos para transmissão dos dados poderiam comprometer a usabilidade do sistema em cenários de votação em larga escala.

Entretanto, não há um valor de referência consolidado na literatura que determine um tempo máximo aceitável para esse tipo de autenticação baseada em QR codes no contexto eleitoral. Dessa forma, este trabalho não estabelece um limite rígido de aceitabilidade, mas busca investigar experimentalmente quais tempos podem ser obtidos utilizando hardware de baixo custo e sem conexão à Internet.

Além do tempo médio de leitura, a utilização de uma câmera de baixa resolução e baixo custo constitui um requisito importante do estudo, uma vez que soluções dependentes de hardware especializado poderiam dificultar a implantação em larga escala em sistemas eleitorais.

4. Método

Nesta seção, detalha-se a metodologia experimental adotada, abrangendo a geração dos QR codes, seu sequenciamento em vídeo e o processo de leitura e decodificação.

4.1. Estruturação dos QR codes

Os QR codes são gráficos em duas dimensões que permitem o armazenamento de informações a partir da codificação em uma matriz de módulos pretos e brancos, na qual a configuração e a quantidade desses módulos determinam a capacidade de armazenamento [Tiwari 2016].

Para a análise e codificação dos dados de entrada, dois parâmetros essenciais à determinação da capacidade de dados são o modo de codificação e o nível de correção de erro estabelecidos. Apesar de os modos numérico e alfanumérico possuírem maiores capacidades de caracteres considerando uma mesma versão de QR code [ISO/IEC 2024], para esse estudo foi necessária a codificação binária dos dados, em que cada caractere ocupa 8 bits. Considerando que as informações a serem armazenadas no QR devem ser cifradas, é preciso ser possível codificar símbolos e caracteres especiais, condições não atendidas pelos outros modos. Quanto à correção de erro, considerou-se o nível M (médio), que estabelece 15% de correção, seguindo o padrão mais usado em aplicações [Tiwari 2016] e garantindo maior confiabilidade de decodificação em caso de partes da imagem do QR code não estarem adequadas para a leitura.

A fim de gerar os QR codes, foi utilizado um protótipo de aplicativo de mobilidade na votação, desenvolvido em React Native, framework de código aberto para desenvolvimento de aplicativos nativos para Android e iOS. Por meio dele, utilizou-se a biblioteca padrão do React Native para QR codes, react-native-qrcode-svg. Para armazenamento do total de informações, os dados foram divididos em diferentes QR codes. Cada um deles tem, em seu cabeçalho, um índice que o identifica como sendo parte de um conjunto maior de códigos. Por exemplo, se um QR code pertencente a um grupo de 35 códigos contiver, após sua decodificação, a informação "idx":12,"total":35 em seu cabeçalho, isso indica que ele é o 12º QR code no total de 35.

A partir do padrão de 12 kB de dados estabelecido para os testes neste estudo, foram gerados QR codes em diferentes versões, correspondendo a distintas divisões de número de QRs, conforme a relação na Tabela 1. Uma vez que o aumento das versões de QR code implica maior densidade de dados em uma única imagem, também se torna mais difícil a sua leitura por dispositivos que não tenham alta resolução de captura.

Tabela 1. Associação entre quantidade de QR codes e versão utilizada.

Quantidade de QR Codes	Versão
17	23
19	22
21	21
23	20
25	19
30	18
35	17
40	16

Com o objetivo de permitir uma leitura rápida dos códigos, com mínima interação tanto do eleitor quanto do mesário, estabeleceu-se o sequenciamento dos QR codes para captura contínua. Assim, foi utilizado um script Python com a biblioteca cv2 para gerar um vídeo com apresentação sequencial. No programa, a partir das imagens de cada conjunto distinto de versões de QRs, parametrizou-se o número desejado de quadros por segundo (FPS - frames per second), automatizando o processo.

4.2. Captura e processamento dos dados

O dispositivo utilizado neste estudo para a captura dos QR codes trata-se de uma câmera digital web equipada com sensor de imagem CMOS, conectada ao computador host por meio de interface USB para transmissão de dados em tempo real. Em relação à taxa de captura, a câmera opera a 30 FPS. O modo de cor adotado é o de cor verdadeira de 24 bits (true color), com tamanho de janela de captura de até 800×600 pixels. Trata-se de um dispositivo de baixo custo, adquirido por aproximadamente R\$ 30,00 (valor de referência em fevereiro de 2026), em contraste com leitores de código de barras bidimensionais dedicados disponíveis no mercado nacional, cujos preços variam entre R\$ 149,00 e R\$ 419,90 — de 5 a 14 vezes mais caros que a câmera utilizada nos testes.

Além disso, utilizou-se outro script Python que teve como propósito realizar a integração entre o funcionamento da câmera e a captura e decodificação das imagens dos códigos. O processo foi realizado de forma concorrente por meio de Python Threads a fim de otimizar o desempenho. A primeira thread foi responsável unicamente pela captura dos frames pela câmera digital web, conectada ao computador por cabo USB. Uma vez capturados, os frames foram adicionados a uma fila. Concomitantemente a essa etapa, a segunda thread foi encarregada de, dentre as imagens presentes na fila, identificar a presença de um QR code e decodificá-lo. Uma vez decodificado, o índice correspondente à posição do código em relação ao todo era armazenado em outra fila. Dessa forma,

apenas informações correspondentes a QR codes ainda não presentes na fila de índices lidos eram enviadas à saída. A execução era interrompida apenas após todos os índices estarem presentes na fila.

4.3. Amostragem e análise de dados

A avaliação do tempo de captura dos dados e decodificação foi feita por meio do armazenamento do intervalo total de cada execução do programa contendo as duas threads de Python. Esse processo de armazenamento foi automatizado a partir do uso da biblioteca openpyxl para integração com um arquivo Excel. Nesse arquivo, foram criadas planilhas para cada uma das combinações de versão do QR code e quadros por segundo de vídeo consideradas nos casos de teste. Ao final de cada execução com leitura completa dos 12 kB de dados, o tempo era adicionado à planilha correspondente. As informações obtidas a partir de 35 repetições de cada caso de teste foram então utilizadas para cálculo de média e desvio padrão a fim de avaliar os resultados.

5. Resultados e discussão

A Figura 1 apresenta a distribuição dos tempos médios de leitura e desvios padrão para cada taxa de reprodução, em FPS, do vídeo usado para captura deles.

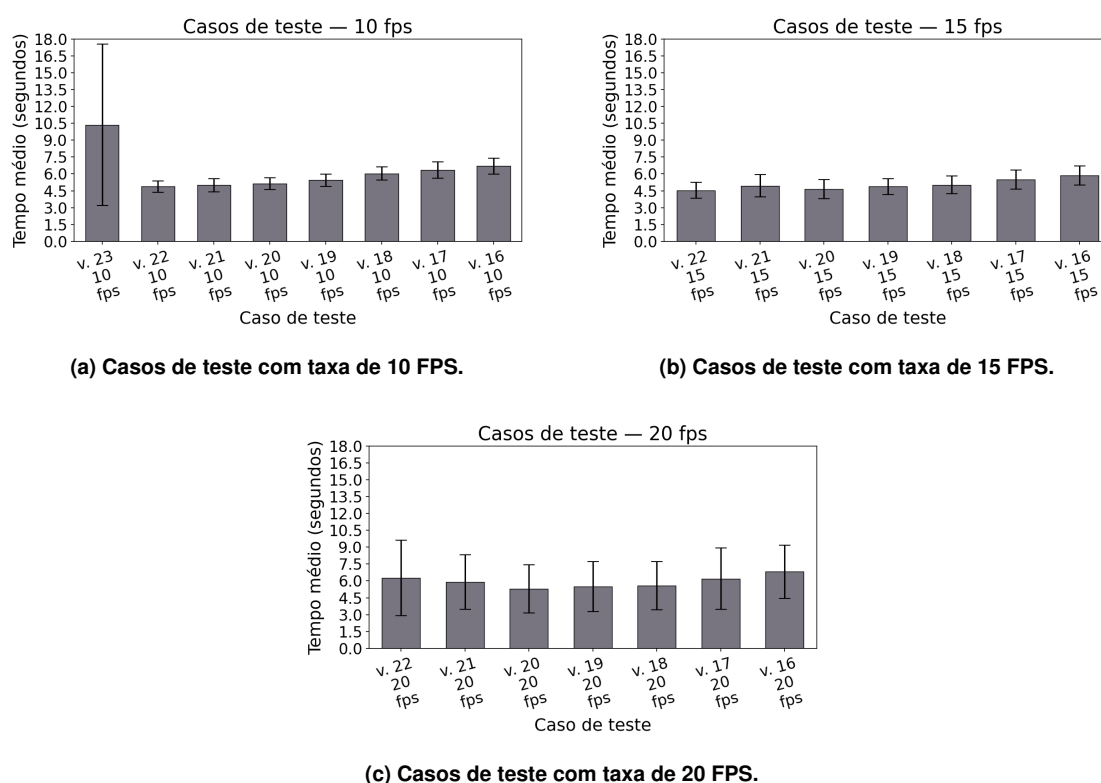


Figura 1. Casos de teste de acordo com a taxa de exibição do vídeo de QR codes.

Nota-se que, mesmo no caso de reprodução mais lenta considerada do vídeo, isto é, 10 FPS, os QR codes de versão 23 apresentaram tanto tempo médio de leitura quanto desvio padrão muito mais elevados do que as demais versões testadas, que possuem menor densidade. A partir disso, inferiu-se que esta versão tem densidade superior à suportada

para decodificação pela resolução da câmera utilizada. Assim, não foram realizados testes com versões superiores à 23, nem da própria versão 23 em maiores taxas de exibição.

Além disso, é possível observar que, principalmente nos casos de leitura a 10 FPS e 15 FPS, ocorreu uma tendência de crescimento do tempo médio de leitura dos QR codes segundo a diminuição da versão. Isso pode ter ocorrido uma vez que um aumento do número total de códigos que compõem a sequência a ser lida resulta em vídeos mais longos, o que também retarda a reexibição completa do conjunto de dados. Observado esse padrão, não foram realizados testes com QR codes de versão inferior a 16.

Já a Figura 2 exibe os casos de teste dos QR codes de versões 22, 19 e 16 segundo o número de quadros por segundo do vídeo.

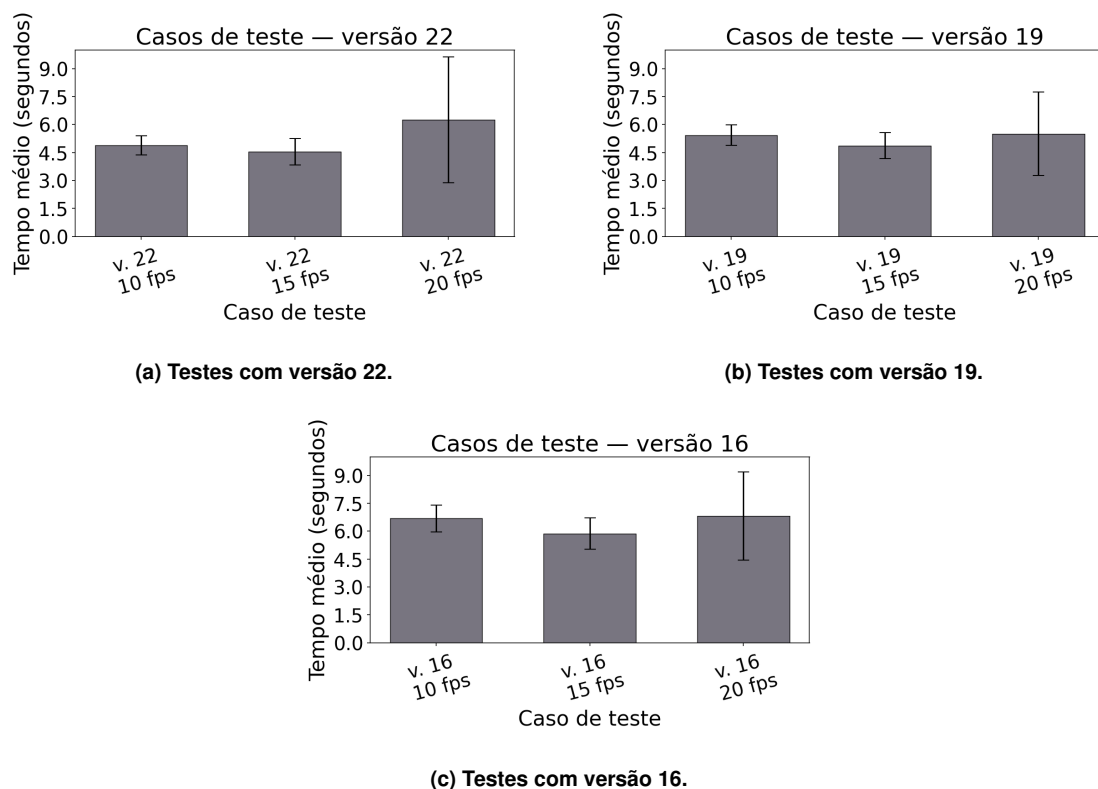


Figura 2. Casos de teste de acordo com a versão dos QR codes.

Analisando os resultados presentes na Figura 2, nota-se que o número de quadros por segundo se relacionou com o desvio padrão de maneira que maiores taxas de reprodução apresentaram maior desvio padrão no tempo de leitura. Isso sugere que, a depender da velocidade de exibição dos quadros do vídeo, a velocidade de captura da câmera pode ser insuficiente para a obtenção de frames com qualidade adequada para decodificação dos QR codes em uma única reprodução.

Verifica-se, nestes testes, que aqueles cujos vídeos foram exibidos a 10 FPS e 15 FPS apresentaram desvios padrão menores. Enquanto isso, os testes com vídeos a 20 fps tiveram desvio padrão em média 3,4 vezes maior do que aqueles a 15 fps. Isso pode decorrer do fato de que na leitura de 30 quadros por segundo, conforme o modelo da câmera utilizada, um vídeo com imagens reproduzidas a 10 FPS tem cada frame capturado 3 vezes pela câmera. Enquanto isso, no caso de um vídeo a 15 FPS, o mesmo frame é

capturado 2 vezes. Para taxas superiores a 15 FPS, cada frame é capturado apenas uma vez. Assim, pode ser avaliado que uma menor quantidade de imagens de cada QR distinto oferece menos amostras para decodificação. Aumenta-se, portanto, a probabilidade de falhas em uma única reprodução do vídeo, exigindo mais exibições para que sejam feitas recapturas e, conseqüentemente, ocasionando aumento do desvio padrão.

A Tabela 2 exibe com maior precisão os tempos médios e desvios padrão de leitura obtidos por versão do QR code e por taxa de exibição do vídeo, incluindo o coeficiente de variação que relaciona cada desvio padrão com sua média correspondente. Além disso, a Figura 3, representa visualmente todos os casos de teste ordenados segundo a ordem crescente de tempo médio de leitura obtido.

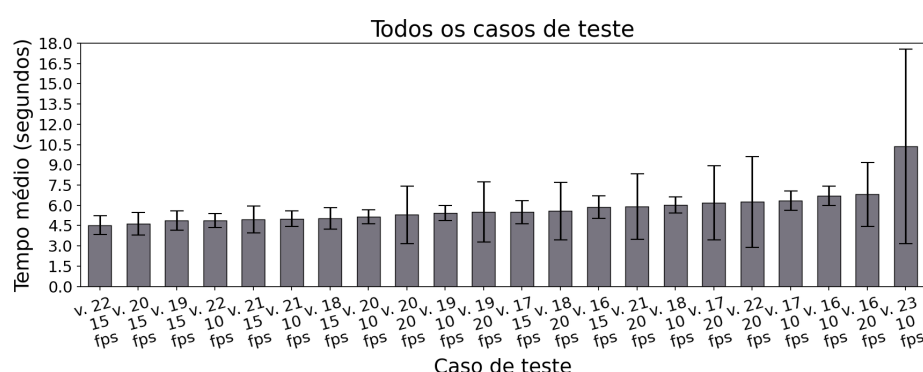


Figura 3. Distribuição do tempo médio de leitura e desvio padrão segundo versão dos QR codes e taxa de exibição, em ordem crescente de tempo médio.

Tabela 2. Tempo médio, desvio padrão e coeficiente de variação de leitura de QR Code por versão e taxa de exibição.

Versão do QR Code	Taxa de Exibição (fps) – Média $\pm$ DP (s)			CV (%)
	10	15	20	
22	4.88 $\pm$ 0.51	4.53 $\pm$ 0.71	6.26 $\pm$ 3.37	10.50 / 15.70 / 53.90
21	5.00 $\pm$ 0.58	4.94 $\pm$ 1.00	5.91 $\pm$ 2.43	11.60 / 20.20 / 41.10
20	5.15 $\pm$ 0.53	4.64 $\pm$ 0.85	5.29 $\pm$ 2.14	10.20 / 18.20 / 40.50
19	5.43 $\pm$ 0.55	4.87 $\pm$ 0.70	6.23 $\pm$ 3.68	10.20 / 14.50 / 59.10
18	6.03 $\pm$ 0.59	5.02 $\pm$ 0.80	5.57 $\pm$ 2.14	9.80 / 15.90 / 38.40
17	6.35 $\pm$ 0.71	5.50 $\pm$ 0.85	6.20 $\pm$ 2.74	11.20 / 15.50 / 44.20
16	6.69 $\pm$ 0.72	5.87 $\pm$ 0.85	6.82 $\pm$ 2.37	10.74 / 10.74 / 34.70

CV = (DP / Média) $\times$ 100. Valores separados por “/” correspondem a cada fps testado, da esquerda para a direita.

Esses dados indicam que o menor tempo médio (4,53 segundos) foi obtido utilizando-se de QRs de versão 22 a 15 quadros por segundo. Porém, a melhor relação entre desvio padrão e tempo médio, conforme demonstrado pelo menor coeficiente de variação (10,20%) corresponde aos casos de QRs de versão 19 e de versão 20, ambos com vídeo a 10 FPS.

A escolha da versão e da taxa de exibição a serem utilizadas em uma aplicação real deve considerar, além de desempenho médio satisfatório, um alto grau de confiabilidade. Neste estudo, a confiabilidade é avaliada no sentido de menores desvios padrão, correspondentes aos vídeos exibidos a 10 e 15 FPS. Já considerando um caso que combina o menor tempo médio de leitura obtido com um desvio padrão baixo, o uso de QR codes de versão 22 exibidos a 15 FPS foi a que apresentou melhor resultado.

A partir das condições estabelecidas na seção 3 (Requisitos), os testes indicam resultados promissores no que tange à usabilidade dos QR codes como ferramenta para transferência rápida de dados. Nesse sentido, visa-se a viabilizar o uso de tokens de votação para validação de dados essenciais à mobilidade na votação.

6. Conclusão

Este estudo investigou a transmissão de grandes volumes de dados sem conexão à Internet por meio de QR codes, visando à viabilização da mobilidade de votação com o uso de tokens. Para isso, foram gerados QR codes de diferentes densidades, sequenciados em vídeos com quadros por segundo distintos, que foram capturados com uma câmera web com conexão USB e decodificados por meio de um programa em Python.

Ao analisar as métricas de tempo médio e desvio padrão obtidas para cada combinação de versão de QR code e de FPS, foi possível notar que maiores velocidades de exibição do vídeo se relacionaram a desvios padrão de tempo de leitura mais elevados. Já considerando o tempo médio, os QR codes que apresentaram melhor desempenho foram os de versões 19 e 20, correspondentes aos com maior densidade de dados entre as versões analisadas. A melhor relação obtida entre desvio padrão e tempo médio ocorreu na leitura de QR codes de versão 19 com vídeo a 15 FPS ($4,87 \pm 0,70$ s).

Comparando-se esses resultados com um critério de referência de até 8 segundos por eleitor — considerado razoável para não comprometer o fluxo de votação —, verifica-se que o uso dos QR codes é viável para a validação das informações do eleitor de forma offline, mesmo em cenários de grande volume, como um local de votação escolhido em período próximo à eleição, propiciando a mobilidade eleitoral.

Destaca-se que este trabalho representa uma etapa inicial de avaliação da viabilidade técnica da proposta, com foco na transmissão rápida de dados; considerações mais aprofundadas sobre segurança — como resistência a ataques específicos (ex: malware no dispositivo móvel, falhas de captação exploradas deliberadamente) e testes com outros dispositivos de captura — são apontadas como direções para trabalhos futuros.

Com relação a outros dispositivos de captura, pode-se considerar a utilização de câmeras com maiores resoluções e taxa de captura. O aprimoramento da resolução contribuiria para a decodificação de QR codes mais densos, enquanto uma taxa de captura superior possibilitaria a leitura de mais códigos por segundo. Além disso, considera-se o uso de dispositivos de captura e decodificação mais especializados, como leitores de código de barras em duas dimensões.

Referências

Chief Election Officer (2015). Administrative report – 2014 municipal election review. Technical report, City of Vancouver. Disponível em: <https://vancouver.ca/f>

iles/cov/administrative-report-2014-municipal-election-review.pdf Acesso em: 25 maio 2026.

Engelbrecht, J., Grüning, M., and Busch, C. (2023). Benchmarking fixed-length fingerprint representations across different embedding sizes and sensor types. *arXiv preprint arXiv:2307.08615*.

Folha de S.Paulo (2022). Datafolha: 22% dos que não votaram no 1º turno citam motivo de saúde. Disponível em: <https://www1.folha.uol.com.br/poder/2022/10/datafolha-22-dos-que-nao-votaram-no-1o-turno-citam-motivo-de-saude.shtml> Acesso em: 25 maio 2026.

Heinrich Böll Stiftung Greece (2024). Decoding the abstention from elections. Disponível em: <https://gr.boell.org/en/2024/10/21/i-aktinografia-tis-apohis-eyrimata-panelladikis-dimoskopisis>. Acesso em: 28 maio 2026.

Heng, Z., Jianguo, H., Haiyan, X., Xiaolong, L., Linfang, Y., and Junpeng, Z. (2021). Research on QR code file transmission technology in complex environment. In *2021 IEEE International Conference on Electronic Technology, Communication and Information (ICETCI)*, pages 16–20.

ISO/IEC (2024). Information technology — automatic identification and data capture techniques — QR code bar code symbology specification. Standard ISO/IEC 18004:2024, International Organization for Standardization.

Kimura, L. and Simplicio Jr., M. (2025). Como implementar “mobilidade na votação” em eleições brasileiras. In *Anais Estendidos do XXV Simpósio Brasileiro de Cibersegurança*, pages 350–354, Porto Alegre, RS, Brasil. SBC.

Mahansaria, D. and Roy, U. K. (2020). Secure authentication using one time contextual QR code. In Thampi, S., Perez, G. M., Ko, R., and Rawat, D., editors, *Security in Computing and Communications (SSCC 2019)*, volume 1208 of *Communications in Computer and Information Science*. Springer, Singapore.

Rizky, M. and Munir, R. (2023). Data transmission to mobile devices via multiple QR codes. In *2023 10th International Conference on Advanced Informatics: Concept, Theory and Application (ICAICTA)*.

Stein, R. M. and Vonnahme, G. (2012). Effect of election day vote centers on voter participation. *Election Law Journal: Rules, Politics, and Policy*.

Tiwari, S. (2016). An introduction to QR code technology. In *Proceedings of the International Conference on Information Technology (ICIT)*, pages 39–44.

TRE-SP (2024). Biometria. Tribunal Regional Eleitoral de São Paulo. Disponível em: <https://www.tre-sp.jus.br/eleicoes/eleicoes-2024/biometria>. Acesso em: 28 maio 2026.

Tribunal Superior Eleitoral (2023). Brasil tem a maior eleição informatizada do mundo. Disponível em: <https://www.tse.jus.br/comunicacao/noticias/2023/Outubro/brasil-tem-a-maior-eleicao-informatizada-do-mundo>. Acesso em: 28 maio 2026.

Tribunal Superior Eleitoral (2025). Comparecimento e abstenção. Disponível em: Portal de Estatísticas do TSE – <https://sig.tse.jus.br/ords/dwapr/r/seai/sig-eleicao-comp-abst/evolucao-de-comp-e-abst>. Acesso em: 25 maio 2026.