

Generador de Códigos QR

Jorge Sánchez
Carlos Guasp



Idea

Generar con simplemente el argumento de una URL una imagen que al ser escaneada con un lector de QR dirija directamente a esta URL.

Lenguajes implementados

- Python
- JavaScript
- Java
- C++

Python

```
def genera_Qr(url, ficheroSalida='codigo_qr.png'):
    qr = qrcode.QRCode(
        version=1,
        error_correction=qrcode.constants.ERROR_CORRECT_L,
        box_size=10,
        border=4,
    )
    qr.add_data(url)
    qr.make(fit=True)

    try:
        validar_url(url)
    except ValueError as e:
        print(f"Error: {e}")
        return

    img = qr.make_image(fill_color="black", back_color="white")

    img.save(ficheroSalida)
    print(f"Código QR guardado en {ficheroSalida}")
```



Java

```
private static void genera_Qr(String Data, int tam, String ficheroSalida)
    throws WriterException, IOException {
    QRCodeWriter qrCodeWriter = new QRCodeWriter();

    BitMatrix bitMatrix = qrCodeWriter.encode(Data,
BarcodeFormat.QR_CODE, tam, tam);

    BufferedImage bufferedImage = new BufferedImage(tam, tam,
BufferedImage.TYPE_INT_RGB);
    for (int i = 0; i < tam; i++) {
        for (int j = 0; j < tam; j++) {
            int color = (bitMatrix.get(i, j)) ? 0xFF000000 : 0xFFFFFFFF;
            bufferedImage.setRGB(i, j, color);
        }
    }

    ImageIO.write(bufferedImage, "png", new File(ficheroSalida));
}
```



JavaScript

```
const QRCode = require('qrcode');

const url = process.argv[2];

if (!url) {
  console.log('Uso: node generar_qr.js <URL>');
  process.exit(1);
}

const ficheroSalida = 'codigo_qr.png';

QRCode.toFile(ficheroSalida, url, {
  width: 350,
  errorCorrectionLevel: 'H',
}, function (err) {
  if (err) throw err;
  console.log(`Código QR guardado en ${ficheroSalida}`);
});
```



C++

```
void genera_Qr(const vector<unsigned char>& qr, int tam, const string&
ficheroSalida) {
    vector<unsigned char> imagen(tam * tam * 4);
    for (int i = 0; i < tam; i++) {
        for (int j = 0; j < tam; j++) {
            int indice = i * tam + j;
            unsigned char color = qr[indice] & 0x01 ? 0 : 255; // negro o blanco
            int indicePixel = 4 * indice;
            imagen[indicePixel] = color;
            imagen[indicePixel + 1] = color;
            imagen[indicePixel + 2] = color;
            imagen[indicePixel + 3] = 255;
        }
    }
    unsigned int error = lodepng::encode(ficheroSalida, imagen, tam, tam);
    if (error) {
        cerr << "Error guardando PNG: " << lodepng_error_text(error) << endl;
    } else {
        cout << "QR guardado en " << ficheroSalida << endl;
    }
}
```

