

Programación de códigos QR en C++

Periféricos y Dispositivos de Interfaz Humana



¿Qué son?

Código QR (Quick Response Code)

- Módulo para almacenar información en una matriz de datos
- Evolución del código de barras (Bidimensional)



12345678900000000000



Características

- Marcadores de posición
- Fácilmente escaneables
- Tolerancia a errores
- Escalabilidad
- Uso (marketing, pagos, eventos...)



Normalmente módulos de 25 x 25

Programación de códigos QR

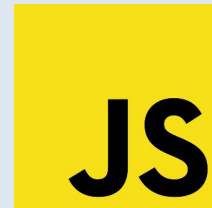
Python

- qrcode
- pyqrcode



JavaScript

- QRCode.js
- jsQR



PHP

- Phpqrcode
- Endroid QRcode



C/C++

- libqrencode
- ZXing-C++





libqrencode

Estructura principal

```
Typedef struct {  
    int version;           // versión del código QR (1-40)  
    int width;             // ancho del código QR en módulos  
    unsigned char *data;   // puntero a array que contiene los datos del QR  
} Qrcode;
```

libqrencode

Generación de códigos QR

/**

*** Crea un objeto QR a partir de la cadena.**

*** @param string Cadena de entrada**

*** @param version Versión del símbolo. Si es 0, la biblioteca elige la versión mínima para los datos de entrada dados.**

*** @param level Nivel de corrección de errores (B,M,Q,H)**

*** @param hint Indica a la biblioteca cómo se deben codificar los caracteres.**

*** @param casesensitive Sensible a mayúsculas y minúsculas (1) o no (0).**

*** @return Una instancia de la clase QRcode.**

***/**

QRcode *QRcode_encodeString(const char *string, int version, QRecLevel level, QRencodeMode hint, int casesensitive);



libqrencode

Liberación de memoria

/**

*** Libera el objeto de entrada.**

*** Todos los fragmentos de datos en el objeto de entrada también se liberan.**

*** @param input Objeto de entrada.**

***/**

void QRinput_free(QRinput *input);

libqrencode

Exportación. No existen funciones en la librería. (Manualmente)

- **ASCII (por pantalla):**

```
void QRcode_print(const QRcode* qrcode) {  
    for (int y = 0; y < qrcode->width; ++y) {  
        for (int x = 0; x < qrcode->width; ++x) {  
            if (qrcode->data[y * qrcode->width + x] & 1) {  
                cout << "■"; // Módulo lleno  
            } else {  
                cout << " "; // Módulo vacío  
            }  
        }  
        cout << endl;  
    }  
}
```

- **PNG:** Es más complejo. Son necesarias otras herramientas como libpng

EJEMPLO

```
#include <qrcode.h>
#include <iostream>
#include <fstream>

using namespace std;

int main() {
    const char* text = "https://www.youtube.com";

    Qrcode* qrcode = Qrcode_encodeString(text, 0, QR_ECLEVEL_L, QR_MODE_8, 1);

    if (!qrcode) {
        cerr << "Error al generar el código QR" << endl;
        return 1;
    }

    ofstream out("qrcode.txt");
    int width = qrcode->width;
    for (int y = 0; y < width; ++y) {
        for (int x = 0; x < width; ++x) {
            if (qrcode->data[y * width + x] & 1) {
                out << "■";
            } else {
                out << " ";
            }
        }
        out << endl;
    }

    Qrcode_free(qrcode);
    cout << "Código QR generado en qrcode.txt" << endl;

    return 0;
}
```

g++ qr.cpp -o qr -lqrencode

www.youtube.com





Otras funciones

- Soporte para archivos SVG y otros formatos gráficos
- Soporte para salida binario y hexadecimal
- Incrustación de imágenes y logotipos



FIN