necesito hacer códigos qr masivo que contenga links a direcciones de internet asociados a elementos de una base de datos tipo "abcde", de tal forma que el link me quede: "https://xxxx.io/?c=abcde". Este qr me gustaría que contenga un logo en el centro. Cual sería la mejor forma de hacerlo sin utilizar herramientas de pago?

ChatGPT dijo:

ChatGPT

Para generar códigos QR en masa con una URL específica y un identificador de elemento único (como el código "abcde" de tu base de datos), y además incluir un logo en el centro, puedes utilizar bibliotecas gratuitas de Python. Aquí te explico el proceso para hacerlo:

Instala las bibliotecas necesarias:

Necesitarás qrcode para generar los códigos QR, Pillow para manipular las imágenes y pandas para manejar tu base de datos (si la tienes en formato CSV, Excel o similar).

bash

Copiar código

pip install qrcode[pil] pillow pandas

Crea un script en Python:

Aquí tienes un script que lee los códigos de una base de datos (por ejemplo, un archivo CSV), genera los códigos QR con la URL y agrega un logo en el centro:

python

Copiar código

import qrcode

from PIL import Image

import pandas as pd

# Cargar el logo

logo\_path = "logo.png" # Asegúrate de tener un logo cuadrado y preferiblemente en formato PNG

logo = Image.open(logo\_path)

# Redimensionar el logo

logo\_size = 80 # Tamaño del logo

logo = logo.resize((logo\_size, logo\_size))

# Leer base de datos (cambia 'data.csv' por el nombre de tu archivo)

df = pd.read\_csv("data.csv") # Asegúrate de tener una columna "code" en el archivo CSV

for index, row in df.iterrows():

code = row["code"]

url = f"https://xxxx.io/?c={code}"

# Generar el QR

qr = qrcode.QRCode(

version=1,

error\_correction=qrcode.constants.ERROR\_CORRECT\_H,

box\_size=10,

border=4,

)

qr.add\_data(url)

qr.make(fit=True)

# Crear imagen QR y agregar logo

qr\_img = qr.make\_image(fill="black", back\_color="white").convert("RGB")

qr\_width, qr\_height = qr\_img.size

pos = ((qr\_width - logo\_size) // 2, (qr\_height - logo\_size) // 2)

qr\_img.paste(logo, pos)

# Guardar la imagen QR

qr\_img.save(f"QR\_{code}.png")

Prepara el logo:

Es recomendable que el logo esté en formato PNG y en tamaño cuadrado (ej., 100x100 píxeles). Si el logo es más grande, asegúrate de reducirlo con el código o manualmente antes de correr el script, ya que si es demasiado grande puede dificultar la legibilidad del QR.

Crea un archivo CSV con los datos:

Debes crear un archivo data.csv que contenga la columna code con los identificadores de cada elemento. Por ejemplo:

css

Copiar código

code

abcde

fghij

klmno

Ejecuta el script:

Esto generará archivos PNG en el mismo directorio, nombrados como QR\_abcde.png, QR\_fghij.png, etc., cada uno con la URL única y el logo en el centro.

Este método es efectivo para una gran cantidad de códigos QR y es completamente gratuito. Además, te da la flexibilidad de modificar los tamaños y colores del QR o del logo según lo necesites.

4o

Dijiste:

nunca he usado python. me podes indicar paso a paso como lo que debo hacer, desde instalar pyton hasta generar el qr por favor?

ChatGPT dijo:

ChatGPT

Memoria actualizada

¡Claro, Gustavo! Aquí tienes una guía paso a paso para instalar Python y ejecutar el código necesario para generar los códigos QR. ¡Vamos a hacerlo desde cero!

Paso 1: Instalar Python

Descarga Python:

Ve a <https://www.python.org/downloads/> y descarga la última versión de Python.

Instala Python:

Abre el archivo descargado e inicia la instalación.

Asegúrate de marcar la casilla que dice **"Add Python to PATH"** antes de hacer clic en **Install Now**. Esto es importante para poder ejecutar Python desde cualquier lugar en tu computadora.

Verifica la instalación:

Abre la **Terminal** en Mac o Linux, o el **Símbolo del sistema (CMD)** en Windows.

Escribe python --version y presiona **Enter**. Si todo salió bien, debería mostrarte la versión de Python instalada.

Paso 2: Instalar las Bibliotecas Necesarias

Abre una terminal o símbolo del sistema:

En Windows, busca "cmd" y abre el Símbolo del sistema.

En Mac, abre la Terminal.

Instala las bibliotecas:

Copia y pega el siguiente comando en la terminal y presiona **Enter**:

bash

Copiar código

pip install qrcode[pil] pillow pandas

Esto instalará las bibliotecas qrcode (para generar los códigos QR), pillow (para manipular imágenes) y pandas (para leer datos desde archivos CSV).

Paso 3: Prepara los Archivos Necesarios

Crea el archivo CSV:

Abre un editor de texto (como Notepad en Windows o TextEdit en Mac) o usa un editor de hojas de cálculo (como Excel).

Crea un archivo llamado data.csv con una lista de tus códigos, en una columna llamada code. Este archivo debe verse así:

css

Copiar código

code

abcde

fghij

klmno

Guarda el archivo en la misma carpeta donde planeas ejecutar el script.

Coloca el logo:

Guarda el logo que quieres en formato PNG en la misma carpeta y asegúrate de que se llame logo.png.

Paso 4: Escribe el Script en Python

Crea el archivo de Python:

Abre el Bloc de notas (Notepad) o cualquier editor de texto y pega el siguiente código en el archivo:

python

Copiar código

import qrcode

from PIL import Image

import pandas as pd

# Cargar el logo

logo\_path = "logo.png" # Asegúrate de tener un logo cuadrado y en formato PNG

logo = Image.open(logo\_path)

# Redimensionar el logo

logo\_size = 80 # Tamaño del logo

logo = logo.resize((logo\_size, logo\_size))

# Leer base de datos

df = pd.read\_csv("data.csv") # Asegúrate de tener una columna "code" en el archivo CSV

for index, row in df.iterrows():

code = row["code"]

url = f"https://xxxx.io/?c={code}"

# Generar el QR

qr = qrcode.QRCode(

version=1,

error\_correction=qrcode.constants.ERROR\_CORRECT\_H,

box\_size=10,

border=4,

)

qr.add\_data(url)

qr.make(fit=True)

# Crear imagen QR y agregar logo

qr\_img = qr.make\_image(fill="black", back\_color="white").convert("RGB")

qr\_width, qr\_height = qr\_img.size

pos = ((qr\_width - logo\_size) // 2, (qr\_height - logo\_size) // 2)

qr\_img.paste(logo, pos)

# Guardar la imagen QR

qr\_img.save(f"QR\_{code}.png")

Guarda el archivo como generador\_qr.py en la misma carpeta donde están data.csv y logo.png.

Paso 5: Ejecuta el Script

**Abre una terminal** en la carpeta donde guardaste los archivos (en Windows puedes hacerlo desde el Explorador de archivos haciendo clic derecho y seleccionando "Abrir en Terminal" o "Abrir en Símbolo del sistema").

Ejecuta el script:

En la terminal, escribe:

bash

Copiar código

python generador\_qr.py

Verifica los archivos generados:

Después de ejecutar el script, verás varios archivos PNG en la misma carpeta. Cada uno de estos archivos será un código QR con el logo en el centro y un enlace único.

¡Y listo! Has generado tus códigos QR personalizados con un logo en el centro.

4o

Dijiste:

que debo modificar para que los qr se almacenen en otra carpeta, por ej: "cuerres"

ChatGPT dijo:

ChatGPT

Para guardar los códigos QR en una carpeta específica, como "cuerres", solo necesitas hacer un pequeño cambio en el script para especificar esa ubicación.

Aquí tienes los pasos:

Crea la carpeta:

Asegúrate de que exista una carpeta llamada "cuerres" en la misma ubicación que tu script. Si no existe, puedes crearla manualmente.

Modifica el código:

Cambia la línea donde se guarda la imagen QR para que incluya la carpeta cuerres/. El código actualizado para esa línea sería:

python

Copiar código

qr\_img.save(f"cuerres/QR\_{code}.png")

Código completo con la modificación:

Aquí tienes el código completo con el cambio realizado:

python

Copiar código

import qrcode

from PIL import Image

import pandas as pd

# Cargar el logo

logo\_path = "logo.png" # Asegúrate de tener un logo cuadrado y en formato PNG

logo = Image.open(logo\_path)

# Redimensionar el logo

logo\_size = 80 # Tamaño del logo

logo = logo.resize((logo\_size, logo\_size))

# Leer base de datos

df = pd.read\_csv("data.csv") # Asegúrate de tener una columna "code" en el archivo CSV

for index, row in df.iterrows():

code = row["code"]

url = f"https://xxxx.io/?c={code}"

# Generar el QR

qr = qrcode.QRCode(

version=1,

error\_correction=qrcode.constants.ERROR\_CORRECT\_H,

box\_size=10,

border=4,

)

qr.add\_data(url)

qr.make(fit=True)

# Crear imagen QR y agregar logo

qr\_img = qr.make\_image(fill="black", back\_color="white").convert("RGB")

qr\_width, qr\_height = qr\_img.size

pos = ((qr\_width - logo\_size) // 2, (qr\_height - logo\_size) // 2)

qr\_img.paste(logo, pos)

# Guardar la imagen QR en la carpeta "cuerres"

qr\_img.save(f"cuerres/QR\_{code}.png")

Ejecuta el script nuevamente:

Después de hacer el cambio, vuelve a ejecutar el script (python generador\_qr.py). Los códigos QR generados se guardarán ahora en la carpeta "cuerres".

4o

Dijiste:

como puedo modificar el tamaño del qr? y la corrección de errores? que otras cosas puedo modificar desde la libreria?

ChatGPT dijo:

ChatGPT

En el código QR puedes ajustar varios parámetros, como el tamaño, la corrección de errores, los colores, e incluso el diseño del borde. Aquí te explico cómo modificar cada uno de estos aspectos utilizando la biblioteca qrcode en Python.

1. Tamaño del Código QR

El tamaño del código QR se controla mediante los parámetros box\_size y border.

**box\_size**: Define el tamaño de cada cuadrado (módulo) en el código QR en píxeles. Aumentar este valor hará que el código QR sea más grande.

**border**: Define el tamaño del borde alrededor del código QR en número de cuadros. Un valor mayor da un borde más grueso.

Por ejemplo, si quieres aumentar el tamaño general del QR:

python

Copiar código

qr = qrcode.QRCode(

version=1,

error\_correction=qrcode.constants.ERROR\_CORRECT\_H,

box\_size=15, # Incrementar el tamaño del cuadro

border=2, # Reducir el borde para que ocupe menos espacio

)

2. Nivel de Corrección de Errores

La corrección de errores es importante para que el código QR sea legible incluso si está parcialmente dañado o si tienes un logo en el centro. La biblioteca qrcode permite 4 niveles de corrección de errores:

**L** (Low) – Corrige hasta el 7% de los datos.

**M** (Medium) – Corrige hasta el 15% de los datos.

**Q** (Quartile) – Corrige hasta el 25% de los datos.

**H** (High) – Corrige hasta el 30% de los datos.

Cuanto más alto sea el nivel de corrección de errores, mayor será la densidad del QR y el tamaño del archivo generado. Para cambiarlo, ajusta el parámetro error\_correction:

python

Copiar código

qr = qrcode.QRCode(

version=1,

error\_correction=qrcode.constants.ERROR\_CORRECT\_Q, # Usa Q para mayor corrección

box\_size=10,

border=4,

)

3. Personalizar el Color del Código QR

La biblioteca qrcode te permite personalizar el color de fondo y el color de los cuadros del código QR mediante el método make\_image:

python

Copiar código

qr\_img = qr.make\_image(fill="blue", back\_color="yellow").convert("RGB")

Aquí fill representa el color de los cuadros del QR, y back\_color representa el color de fondo.

4. Controlar la Densidad del QR con version

El parámetro version controla la densidad del QR. Va de 1 a 40, donde 1 representa un código QR simple y 40 un código muy denso que puede almacenar mucha información. Si version=1 no alcanza para almacenar tu URL (por ejemplo, si tienes muchos caracteres), puedes aumentarlo o establecerlo en None para que ajuste automáticamente.

python

Copiar código

qr = qrcode.QRCode(

version=5, # Aumenta para mayor densidad si es necesario

error\_correction=qrcode.constants.ERROR\_CORRECT\_H,

box\_size=10,

border=4,

)

Ejemplo Completo con Varias Personalizaciones

Aquí tienes un ejemplo que incluye todas estas opciones:

python

Copiar código

import qrcode

from PIL import Image

import pandas as pd

# Cargar el logo

logo\_path = "logo.png"

logo = Image.open(logo\_path)

# Redimensionar el logo

logo\_size = 80

logo = logo.resize((logo\_size, logo\_size))

# Leer base de datos

df = pd.read\_csv("data.csv")

for index, row in df.iterrows():

code = row["code"]

url = f"https://xxxx.io/?c={code}"

# Generar el QR con personalizaciones

qr = qrcode.QRCode(

version=2, # Versión para controlar la densidad

error\_correction=qrcode.constants.ERROR\_CORRECT\_Q, # Corrección media-alta

box\_size=12, # Tamaño del cuadro

border=2, # Tamaño del borde

)

qr.add\_data(url)

qr.make(fit=True)

# Crear imagen QR con color personalizado

qr\_img = qr.make\_image(fill="black", back\_color="white").convert("RGB")

qr\_width, qr\_height = qr\_img.size

pos = ((qr\_width - logo\_size) // 2, (qr\_height - logo\_size) // 2)

qr\_img.paste(logo, pos)

# Guardar la imagen QR en la carpeta especificada

qr\_img.save(f"cuerres/QR\_{code}.png")

Resumen de lo que puedes modificar

**box\_size**: Cambia el tamaño de cada cuadro (aumenta para un QR más grande).

**border**: Ajusta el borde del QR.

**error\_correction**: Define la corrección de errores (usualmente ERROR\_CORRECT\_H o ERROR\_CORRECT\_Q si tienes un logo).

**version**: Controla la densidad del QR; valores mayores permiten más datos.

**fill y back\_color**: Cambian los colores del QR.

Con estas opciones, puedes personalizar tu código QR para que se vea y funcione exactamente como lo necesitas.

4o

Dijiste:

se podría agregar como texto debajo del qr en tamaño chico el "code" de cada qr?

ChatGPT dijo:

ChatGPT

Sí, puedes agregar el código como texto debajo de cada código QR utilizando la biblioteca Pillow para superponer texto en la imagen generada. Aquí te muestro cómo hacerlo en el script.

Modificación del Código

Para agregar el texto, usaremos ImageDraw y ImageFont de Pillow, que nos permiten escribir texto en las imágenes. A continuación, ajustamos el tamaño de la imagen para añadir espacio debajo del código QR y colocamos el texto en ese espacio.

Aquí tienes el código modificado:

python

Copiar código

import qrcode

from PIL import Image, ImageDraw, ImageFont

import pandas as pd

# Cargar el logo

logo\_path = "logo.png"

logo = Image.open(logo\_path)

# Redimensionar el logo

logo\_size = 80

logo = logo.resize((logo\_size, logo\_size))

# Leer base de datos

df = pd.read\_csv("data.csv")

# Configuración de fuente para el texto (ajusta el tamaño de fuente según sea necesario)

try:

font = ImageFont.truetype("arial.ttf", 14) # Usar Arial (asegúrate de tener Arial o cambia a otra fuente disponible)

except IOError:

font = ImageFont.load\_default() # Cargar fuente predeterminada si Arial no está disponible

for index, row in df.iterrows():

code = row["code"]

url = f"https://xxxx.io/?c={code}"

# Generar el QR con personalizaciones

qr = qrcode.QRCode(

version=2,

error\_correction=qrcode.constants.ERROR\_CORRECT\_Q,

box\_size=12,

border=2,

)

qr.add\_data(url)

qr.make(fit=True)

# Crear imagen QR con color personalizado

qr\_img = qr.make\_image(fill="black", back\_color="white").convert("RGB")

# Calcular posición y añadir logo al QR

qr\_width, qr\_height = qr\_img.size

pos = ((qr\_width - logo\_size) // 2, (qr\_height - logo\_size) // 2)

qr\_img.paste(logo, pos)

# Ampliar la imagen para dejar espacio para el texto debajo del QR

new\_height = qr\_height + 30 # Aumenta 30 píxeles para el texto

qr\_with\_text = Image.new("RGB", (qr\_width, new\_height), "white")

qr\_with\_text.paste(qr\_img, (0, 0))

# Escribir el código debajo del QR

draw = ImageDraw.Draw(qr\_with\_text)

text\_width, text\_height = draw.textsize(code, font=font)

text\_x = (qr\_width - text\_width) // 2

text\_y = qr\_height + 5 # Coloca el texto un poco debajo del QR

draw.text((text\_x, text\_y), code, font=font, fill="black")

# Guardar la imagen QR con el texto en la carpeta especificada

qr\_with\_text.save(f"cuerres/QR\_{code}.png")

Explicación de los Cambios

Espacio para el Texto:

Creamos una nueva imagen (qr\_with\_text) con la altura incrementada en 30 píxeles para incluir el texto debajo del código QR.

Añadir el Texto:

Usamos ImageDraw.Draw() para escribir el código (code) en el nuevo espacio debajo del QR.

Centramos el texto debajo del QR usando el ancho calculado (text\_width y text\_x).

Fuente del Texto:

Utilizamos la fuente Arial en tamaño 14 para que el texto sea pequeño y legible. Si Arial no está disponible, se cargará una fuente predeterminada.

Resultado

Al ejecutar este código, cada archivo generado en la carpeta "cuerres" tendrá el código QR con el code correspondiente en texto pequeño debajo del QR.

4o

Dijiste:

me da error: Traceback (most recent call last): File "D:\Dropbox\KIBBO\generador\_qr\_txt.py", line 52, in <module> text\_width, text\_height = draw.textsize(code, font=font) ^^^^^^^^^^^^^ AttributeError: 'ImageDraw' object has no attribute 'textsize'

ChatGPT dijo:

ChatGPT

Ese error ocurre porque la función textsize ya no se usa en las versiones más recientes de Pillow. Ahora se usa textbbox para obtener el tamaño del texto.

Aquí está el código corregido usando textbbox:

python

Copiar código

import qrcode

from PIL import Image, ImageDraw, ImageFont

import pandas as pd

# Cargar el logo

logo\_path = "logo.png"

logo = Image.open(logo\_path)

# Redimensionar el logo

logo\_size = 80

logo = logo.resize((logo\_size, logo\_size))

# Leer base de datos

df = pd.read\_csv("data.csv")

# Configuración de fuente para el texto (ajusta el tamaño de fuente según sea necesario)

try:

font = ImageFont.truetype("arial.ttf", 14) # Usar Arial (asegúrate de tener Arial o cambia a otra fuente disponible)

except IOError:

font = ImageFont.load\_default() # Cargar fuente predeterminada si Arial no está disponible

for index, row in df.iterrows():

code = row["code"]

url = f"https://xxxx.io/?c={code}"

# Generar el QR con personalizaciones

qr = qrcode.QRCode(

version=2,

error\_correction=qrcode.constants.ERROR\_CORRECT\_Q,

box\_size=12,

border=2,

)

qr.add\_data(url)

qr.make(fit=True)

# Crear imagen QR con color personalizado

qr\_img = qr.make\_image(fill="black", back\_color="white").convert("RGB")

# Calcular posición y añadir logo al QR

qr\_width, qr\_height = qr\_img.size

pos = ((qr\_width - logo\_size) // 2, (qr\_height - logo\_size) // 2)

qr\_img.paste(logo, pos)

# Ampliar la imagen para dejar espacio para el texto debajo del QR

new\_height = qr\_height + 30 # Aumenta 30 píxeles para el texto

qr\_with\_text = Image.new("RGB", (qr\_width, new\_height), "white")

qr\_with\_text.paste(qr\_img, (0, 0))

# Escribir el código debajo del QR

draw = ImageDraw.Draw(qr\_with\_text)

text\_bbox = draw.textbbox((0, 0), code, font=font)

text\_width = text\_bbox[2] - text\_bbox[0]

text\_x = (qr\_width - text\_width) // 2

text\_y = qr\_height + 5 # Coloca el texto un poco debajo del QR

draw.text((text\_x, text\_y), code, font=font, fill="black")

# Guardar la imagen QR con el texto en la carpeta especificada

qr\_with\_text.save(f"cuerres/QR\_{code}.png")

Explicación del Cambio

**textbbox**: La función textbbox devuelve las coordenadas de un cuadro delimitador para el texto, lo que permite calcular el ancho y alto.

Calculamos el ancho del texto restando las coordenadas text\_bbox[2] y text\_bbox[0].

Prueba con este código y deberías obtener el texto centrado debajo del código QR sin errores.

4o

Dijiste:

para hacer la letra más grande debo cambiar el "14" en esta línea: "ImageFont.truetype("arial.ttf", 14)"

ChatGPT dijo:

ChatGPT

¡Exactamente! Para aumentar el tamaño de la letra, cambia el valor 14 en la línea:

python

Copiar código

font = ImageFont.truetype("arial.ttf", 14)

Por ejemplo, si quieres que la letra sea más grande, podrías probar con:

python

Copiar código

font = ImageFont.truetype("arial.ttf", 20)

El número representa el tamaño de la fuente en puntos. Solo asegúrate de ajustar el espacio debajo del código QR (es decir, el valor new\_height), si es necesario, para que el texto más grande quepa bien.

4o

Dijiste:

y si quiero que el texto me quede mas cerca del qr

ChatGPT dijo:

ChatGPT

Para acercar el texto al QR, debes reducir el valor de text\_y, que controla la posición vertical del texto en relación con la parte inferior del código QR.

Ajuste del Código

En esta línea:

python

Copiar código

text\_y = qr\_height + 5 # Coloca el texto un poco debajo del QR

Cambia 5 por un valor menor (incluso 0, si quieres que esté justo debajo). Por ejemplo, si pones:

python

Copiar código

text\_y = qr\_height + 2

El texto quedará más cerca del QR. Prueba diferentes valores hasta que logres la distancia que te guste.

4o

Dijiste:

se podría hacer con un logo redondo?

ChatGPT dijo:

ChatGPT

Sí, se puede usar un logo redondo en el centro del código QR. Para ello, solo necesitas asegurarte de que el logo tenga forma circular antes de insertarlo. Esto se puede lograr creando una máscara circular en Pillow para recortar el logo en forma de círculo.

Aquí te muestro cómo hacerlo:

Modificación del Código para un Logo Redondo

Añadiremos una máscara circular para recortar el logo antes de colocarlo en el QR.

python

Copiar código

import qrcode

from PIL import Image, ImageDraw, ImageFont

import pandas as pd

# Cargar el logo y redimensionarlo

logo\_path = "logo.png"

logo = Image.open(logo\_path)

logo\_size = 80

logo = logo.resize((logo\_size, logo\_size))

# Crear una máscara circular para el logo

mask = Image.new("L", (logo\_size, logo\_size), 0)

draw\_mask = ImageDraw.Draw(mask)

draw\_mask.ellipse((0, 0, logo\_size, logo\_size), fill=255)

logo.putalpha(mask)

# Leer base de datos

df = pd.read\_csv("data.csv")

# Configuración de fuente para el texto (ajusta el tamaño de fuente según sea necesario)

try:

font = ImageFont.truetype("arial.ttf", 14) # Usar Arial

except IOError:

font = ImageFont.load\_default() # Cargar fuente predeterminada si Arial no está disponible

for index, row in df.iterrows():

code = row["code"]

url = f"https://xxxx.io/?c={code}"

# Generar el QR con personalizaciones

qr = qrcode.QRCode(

version=2,

error\_correction=qrcode.constants.ERROR\_CORRECT\_Q,

box\_size=12,

border=2,

)

qr.add\_data(url)

qr.make(fit=True)

# Crear imagen QR con color personalizado

qr\_img = qr.make\_image(fill="black", back\_color="white").convert("RGBA")

# Calcular posición y añadir logo redondo al QR

qr\_width, qr\_height = qr\_img.size

pos = ((qr\_width - logo\_size) // 2, (qr\_height - logo\_size) // 2)

qr\_img.paste(logo, pos, mask=logo) # Usar la máscara para insertar el logo redondo

# Ampliar la imagen para dejar espacio para el texto debajo del QR

new\_height = qr\_height + 30 # Aumenta 30 píxeles para el texto

qr\_with\_text = Image.new("RGB", (qr\_width, new\_height), "white")

qr\_with\_text.paste(qr\_img, (0, 0))

# Escribir el código debajo del QR

draw = ImageDraw.Draw(qr\_with\_text)

text\_bbox = draw.textbbox((0, 0), code, font=font)

text\_width = text\_bbox[2] - text\_bbox[0]

text\_x = (qr\_width - text\_width) // 2

text\_y = qr\_height + 2 # Ajustar para acercar el texto al QR

draw.text((text\_x, text\_y), code, font=font, fill="black")

# Guardar la imagen QR con el texto en la carpeta especificada

qr\_with\_text.save(f"cuerres/QR\_{code}.png")

Explicación de los Cambios

Crear una máscara circular:

Creamos una nueva imagen en escala de grises (L) llamada mask, del mismo tamaño que el logo.

Luego, usamos draw\_mask.ellipse() para dibujar un círculo blanco (255) en la máscara.

Aplicar la máscara:

Al usar logo.putalpha(mask), el logo se recorta en forma circular.

Al pegar el logo en el código QR (qr\_img.paste(logo, pos, mask=logo)), solo se inserta la parte circular.

Este código colocará un logo redondo en el centro del QR, haciendo que el diseño sea más uniforme y atractivo.

4o

Dijiste:

ok. mañana lo pruebo

ChatGPT dijo:

ChatGPT

Memoria actualizada

¡Perfecto, Gustavo! Cualquier duda que tengas cuando lo pruebes, no dudes en decirme, estaré aquí para ayudarte. ¡Éxito con los códigos QR!

4o

Dijiste:

me podrías aislar el código que genera el logo circular? de esta manera usaría ese para redondear el logo y el otro para aplicarlo

ChatGPT dijo:

ChatGPT

Claro, aquí tienes el código aislado que crea un logo circular a partir de una imagen cuadrada. Puedes usar este fragmento por separado para generar el logo redondeado y luego guardarlo o usarlo en el script principal.

Código para Generar un Logo Circular

python

Copiar código

from PIL import Image, ImageDraw

# Cargar el logo y redimensionarlo

logo\_path = "logo.png" # Ruta de la imagen original del logo

logo\_size = 80 # Tamaño deseado para el logo

logo = Image.open(logo\_path).resize((logo\_size, logo\_size))

# Crear una máscara circular para el logo

mask = Image.new("L", (logo\_size, logo\_size), 0)

draw\_mask = ImageDraw.Draw(mask)

draw\_mask.ellipse((0, 0, logo\_size, logo\_size), fill=255)

# Aplicar la máscara circular al logo

logo.putalpha(mask)

# Guardar el logo circular en un archivo (opcional)

logo.save("logo\_circular.png") # Guarda el logo con transparencia circular

Explicación de los Pasos

Cargar y Redimensionar el Logo:

Carga el logo desde el archivo y lo redimensiona al tamaño deseado (logo\_size).

Crear la Máscara Circular:

Genera una máscara en escala de grises ("L") del mismo tamaño que el logo.

Dibuja un círculo blanco (255) en la máscara para definir la parte visible.

Aplicar la Máscara:

Aplica la máscara al logo con putalpha(mask), lo que recorta el logo en forma circular.

Guardar el Logo Circular (opcional):

Si deseas usar el logo circular en otro script, puedes guardarlo como "logo\_circular.png".

Este código te permitirá redondear el logo una sola vez y luego usar el archivo logo\_circular.png en tu otro código para generar los QR.

4o

Dijiste:

el qr luce exactamente igual con el logo circular, como si nada hubiera sucedido

ChatGPT dijo:

ChatGPT

Si el logo circular no se nota en el QR, es posible que la transparencia de la imagen no esté aplicándose correctamente, o que la máscara no se esté interpretando al insertarla en el QR.

Probemos con una pequeña modificación para asegurarnos de que el fondo fuera del círculo sea transparente:

Asegurémonos de que el logo circular tenga fondo transparente.

**Guardémoslo en formato PNG** para que se mantenga la transparencia.

Aquí tienes una versión revisada:

python

Copiar código

from PIL import Image, ImageDraw

# Cargar y redimensionar el logo

logo\_path = "logo.png" # Ruta de la imagen original del logo

logo\_size = 80 # Tamaño deseado para el logo

logo = Image.open(logo\_path).resize((logo\_size, logo\_size)).convert("RGBA")

# Crear una máscara circular para el logo

mask = Image.new("L", (logo\_size, logo\_size), 0)

draw\_mask = ImageDraw.Draw(mask)

draw\_mask.ellipse((0, 0, logo\_size, logo\_size), fill=255)

# Crear una imagen circular con fondo transparente

circular\_logo = Image.new("RGBA", (logo\_size, logo\_size))

circular\_logo.paste(logo, (0, 0), mask=mask)

# Guardar el logo circular en un archivo (opcional)

circular\_logo.save("logo\_circular.png") # Guarda el logo con transparencia circular

Explicación de los Cambios

Convertir el Logo a "RGBA":

Al usar convert("RGBA") en el logo original, nos aseguramos de que tenga un canal de transparencia.

Crear una Imagen Transparente:

circular\_logo es una nueva imagen con transparencia (RGBA), que permite que solo el área circular del logo se mantenga visible.

Aplicar la Máscara Circular al Logo:

Usamos paste con mask=mask para pegar el logo dentro de circular\_logo, aplicando la máscara para obtener el círculo.

Guarda el logo circular resultante como logo\_circular.png y úsalo en tu script principal. Con estos ajustes, deberías notar la diferencia visual en el QR.