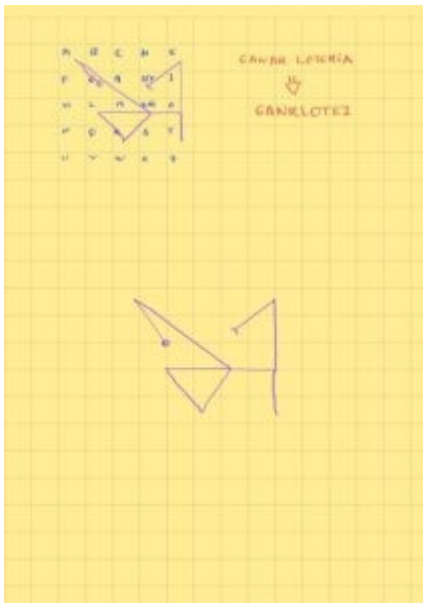


Un sigilo es un glifo pictórico que oculta una declaración o deseo. Os dejo un esquema de como se construyen a partir de un cuadrado mágico y siguiendo una grafía regular.



El proceso sería el siguiente:

1. Se formula un deseo. Por ejemplo "GANAR LOTERÍA"
2. Se quitan los espacios y las letras repetidas: "GANRLOTEI"
3. Las Y las transformamos en I y las Ñ en N.
4. Se construye el cuadrado mágico de 5 x 5 con las letras del alfabeto español. Como el alfabeto español tiene 27 caracteres metemos la I con la Y , y la N con la Ñ; por similitud. Así cuadra.
5. Empezamos por la primera letra, En este caso la G y vamos trazando líneas rectas para unir todas las letras, escogiendo como punto de unión el centro de las celdillas de los cuadrados mágicos, según se observa en la imagen.
6. Una vez hecho el primero, lo aprendemos de memoria y aprendemos a dibujarlo sin ayuda del cuadrado mágico.
7. Luego lo dibujamos en todos los sitios que queramos. Se supone que estos sigilos hay que "cargarlos" mágicamente y que su efectividad radica en los procesos mentales que ocurren cuando tenemos la atención fijada en su construcción dentro del cuadrado mágico.

QB64

Bien, nosotros nos vamos a ocupar de construir una aplicación que dibuje estos sigilos en la pantalla del ordenador. Y para esto ¿qué necesito?.

Pues obviamente, un lenguaje de programación que sea capaz de acceder a la pantalla gráfica. Es decir que tenga funciones para operar a nivel del píxel y capaz de dibujar formas geométricas. Al menos, las mas elementales: líneas, círculos, rectángulos, etc.

Prácticamente todos los lenguajes son capaces de acceder a la pantalla gráfica. Pero el que yo uso mas habitualmente: VBA para Access, Oh sorpresa, no accede a la pantalla gráfica, salvo si se instalan ciertas librerías, que estuve mirando y no había una documentación clara para hacerlo rápido. Simplemente decían: "instálense las librerías" (ya, pero como) y "utilícense las funciones" (si, si pero al menos un ejemplo ¿no?).

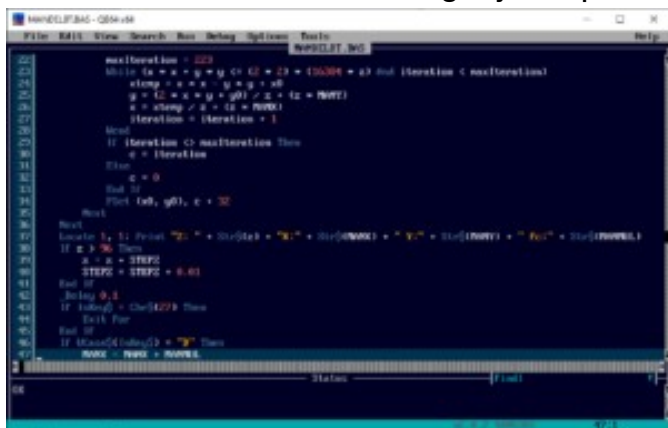
Luego estuve mirando los lenguajes modernillos tipo C# o JAVA y tres cuartos de lo mismo. Te remiten a interminables librerías. Es decir, al final hay que hacer un arco de iglesia para trazar unos simples segmentos.

Y he aquí que apareció QB64, quick basic de 64 bits. Un lenguaje de los de antes, pero modernizado. Y lo que yo buscaba, una colección muy amplia de funciones que hacen todo lo que yo quería y muchas mas cosas. Y sin librerías ni nada. Simplemente forman parte del lenguaje.

Además tiene integración con Visual Studio Code, y lo que es mejor, es un compilador! a la antigua. Que genera archivos EXE stand alone totalmente autónomos.

Una joya. Así que me dispuse a instalarlo y probarlo. Primera sorpresa. No hace falta instalación.

El compilador se puede descargar [aquí](#). Y viene ya con un IDE anticuado pero muy funcional. La sintaxis del lenguaje se puede consultar [aquí](#).



Y [ésta es la página](#) de GitHub de QB64. En ésta web hay una buena entrada sobre QB64.

Instalación De QB64

Pues otra grata sorpresa de QB64. No necesita instalación. Simplemente descargamos el lenguaje que viene en este zip:

qb64_dev_2022-09-08-07-14-00_47f5044_win-x64.7z

Si lo abrimos observaremos que trae una carpeta que se llama qb64. Es tan simple como descomprimir esa carpeta en el lugar que queramos. Por comodidad y mejor funcionamiento yo la he puesto colgando de C:

C:\qb64

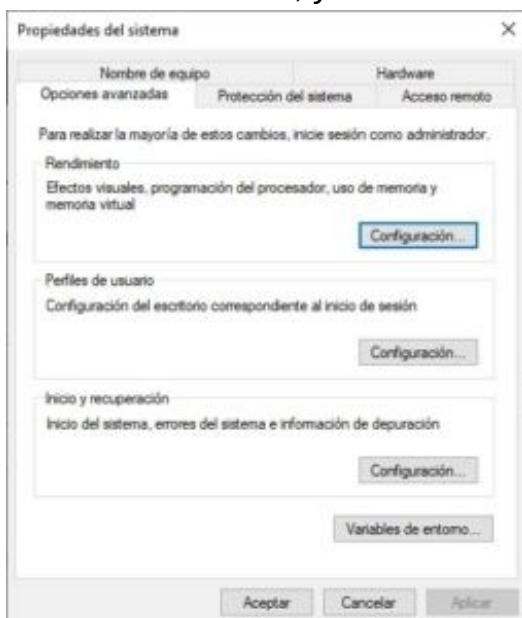
Pero podéis ponerla donde queráis. Eso sí, hay que modificar la variable del sistema: PATH, para añadir la nueva carpeta. Eso se hace abriendo el panel de control:



Pulsamos Sistema y seguridad:

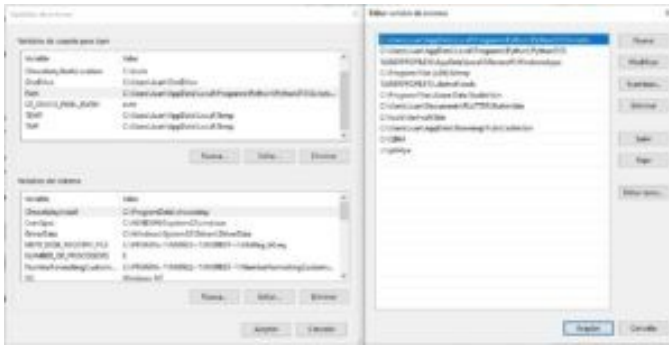


Pulsamos Sistema, y dentro de sistema, Configuración Avanzada del Sistema:



Pulsamos Variables de entorno:

Y dentro de variables de entorno nos posicionamos en: Path y pulsamos editar:



Después pulsamos "nueva" y añadimos: c:\qb64

Salimos grabando y ya tenemos operativa la nueva variable de entorno.

Bueno, pues con esto estaría la instalación del lenguaje. Como habéis visto ni 5 minutos. Se tarda mas en explicarlo que en hacerlo.

Manera De Programar En QB64

Nuevamente, simplicidad extrema. Abrimos una carpeta en cualquier sitio y vamos metiendo ahí todos los archivos relacionados con el proyecto. En mi caso, por simplicidad y por tenerlo todo en un mismo sitio, he abierto la carpeta: SIGILOS en: C:\qb64\SIGILOS

para compilar el programa de forma rápida, he creado un archivo:

e.bat

que contiene:

cls

del /Q sigilos.exe

qb64 -c sigilos.bas -o C:\qb64\sigilos\sigilos.exe

C:\qb64\sigilos\sigilos.exe

Como podéis ver lo que hace es compilar el código fuente y ejecutar el EXE.

El código fuente de QB64 son archivos de texto acabados en .BAS

Para editar esos archivos de texto, se puede usar cualquier editor, pero como somos programadores profesionales lo suyo es usar el Visual Studio Code. A ver, existe una extensión de VSCode para QB64, que me ha funcionado bien en Windows 11, pero Windows 10 no ha habido manera. Esa extensión se puede descargar [aquí](#). Si alguien consigue que funcione en Windows 10, que me explique como.

Bueno, pues ya tenemos el compilador instalado, el entorno de programación configurado (Visual Studio Code), y ahora simplemente arrancamos el VSCode:



En el VSCode ya tengo cargada la carpeta de trabajo:

C:\qb64\SIGILOS

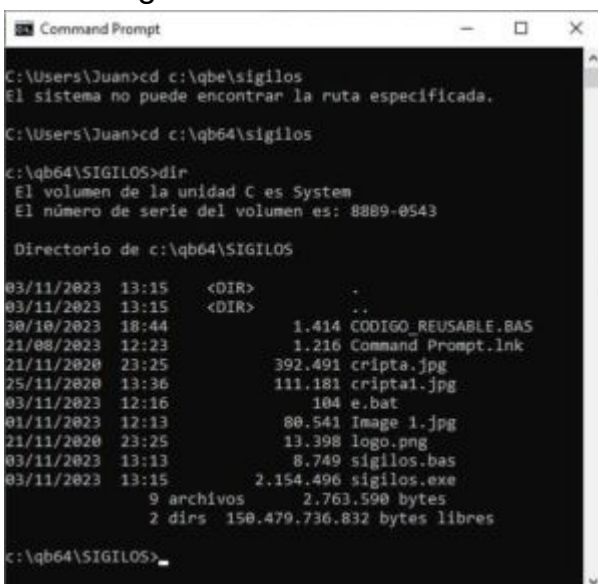
Que ya contiene el fichero de código fuente que ejecuta el programa:

C:\qb64\SIGILOS\sigilos.bas

En el apéndice final se puede ver ese código fuente, que aún lo estoy puliendo pero que ya funciona en lo que debe hacer el programa que es dibujar los sigilos. Faltan toda bastantes detalles, sobre todo de adorno.

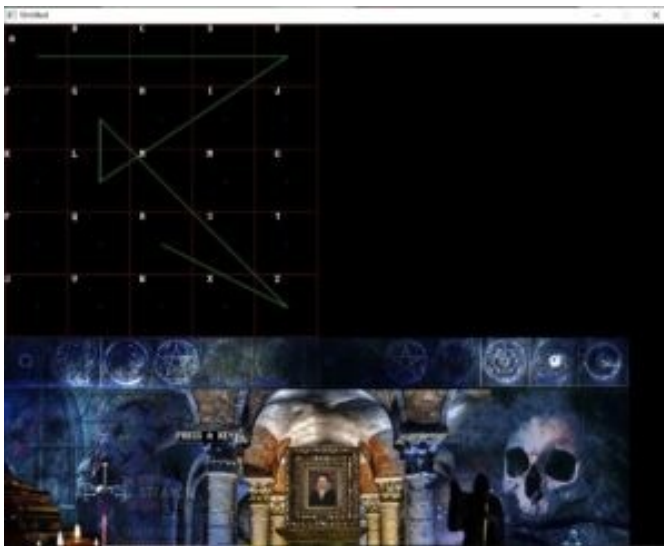
La manera de trabajar es la siguiente:

1. Abro VSCode y voy programando la aplicación en el código fuente.
2. Cuando quiero probar como va mi trabajo y ejecutar la aplicación, lo que hago es abrir un terminal MSDOS.



Y desde este terminal ejecuto:

e.bat



Como veis se reproduce exactamente el sigilo pedido. A la aplicación le faltan cosas, pero lo principal lo hace que era lo que quería mostrar.