

INDICE

	Pág.
1. Introducción	
1.1. Motivación	2
1.2. Objetivo	2
2. Estructura interna de un archivo *.BMP	
2.1. Conceptos básicos	3
2.2. Cabecera del archivo	3
2.3. Cabecera de información	4
2.4. Paleta de colores	4
2.5. Datos de imagen	4
3. Introducción de texto dentro de una imagen BMP	
3.1. Codificación de caracteres en los píxeles de la imagen	5
4. Software	
4.1. Lenguaje de programación empleado	6
4.2. Principios de codificación del texto	6
4.3. Obtención de los bytes de la imagen	
4.3.1. Funciones get & put	7
4.4. Codificación del texto en la imagen	
4.4.1. Funciones lógicas and y or a nivel de bit	8
4.4.2. Codificación de caracteres en la imagen	10
4.4.3. Decodificación de caracteres en la imagen	11
5. Guía de usuario	
5.1. Requisitos mínimos	12
5.2. Instalación	12
5.3. Entorno	15
5.4. Encriptar texto	15
5.5. Desencriptar texto	19
5.6. Desinstalación	22
6. Anexos	
6.1. Código completo de la aplicación	23

1.- Introducción.

1.1.- Motivación.

Como Proyecto Integrado de la titulación de “Técnico Superior de Sistemas de Telecomunicaciones e Informáticos” para la obtención de su título correspondiente y unido a la curiosidad tecnológica sobre una alternativa a la transmisión de mensajes seguros surge este estudio sobre las posibilidades que presenta este sistema de estenografía y su posible aplicación para la transmisión de mensajes de forma segura.

1.2.- Objetivo.

El objetivo de este proyecto es realizar un estudio, mediante una aplicación practica, de la capacidad de ofuscar texto dentro de imágenes para asegurar de este modo la transferencia segura de mensajes con cierta importancia. En el presente proyecto se ha limitado a la ofuscación directa de texto plano, aunque es posible realizarlo sobre texto encriptado con anterioridad mediante algún algoritmo de cifrado, de tal modo que se asegurase aun mas la transmisión de dicho mensaje.

Una de las utilidades de el presente proyecto es la posibilidad de enviar texto usando de base una imagen de tal modo que se asegure su privacidad.

Otra de las posibles utilidades, y quizás más viable es la posibilidad de introducir de este modo el copyright de las imágenes, de tal modo que nadie pueda eliminar, puesto que al ser un pequeño texto, puede ser introducido a partir del píxel que se desee, y por lo tanto sería muy complicado su modificación.

2.- Estructura interna de un archivo *.BMP.

2.1.- Conceptos básicos.

El formato BMP (Bit Map Picture) guarda en el archivo la información de la imagen píxel por píxel. Debido a esta peculiaridad este formato ocupa mucho mas tamaño que otros formatos como podrían ser JPG.

La profundidad de color en este formato puede variar, por norma general pueden ser de 1,4,8 o 24 bits por píxel. Esto indica la cantidad de bits necesarios para poder reproducir un píxel por pantalla, y tiene su correspondencia en colores, como se indica en la siguiente tabla:

BITS	COLORES
1	2 colores
4	16 colores
8	256 colores
24	16,7 millones (Color Real)

En el formato BMP, los píxeles se comienzan a representar por la esquina inferior izquierda de la imagen, de forma totalmente contraria a como se representan imágenes en una pantalla.

En el presente proyecto, solo utilizaremos imágenes de 24 bits, puesto el resto en la actualidad no se usan, y además, estas imágenes nos permiten introducir mas cantidad de datos en la misma imagen y cualquier cambio de tonalidad en un píxel de la imagen pasaría desapercibido al ojo humano, ante 16,7 millones de colores.

2.2.- Cabecera del archivo.

Tamaño	Contenido	Descripción
2 bytes	Tipo de formato	En el formato BMP, estos dos bytes corresponden a los caracteres "BM"
4 bytes	Tamaño del archivo	Tamaño en bytes del archivo.
2 bytes	Reservado	En el formato BMP siempre es 0
2 bytes	Reservado	En el formato BMP siempre es 0
4 bytes	Offset	Posición en bytes donde comienzan los datos de la imagen en sí.

2.3.- Cabecera de información.

Tamaño	Contenido	Descripción
4 bytes	Tamaño de la cabecera	Tamaño en bytes de la cabecera de información. Su valor es siempre 40.
4 bytes	Ancho	El ancho en píxeles de la imagen.
4 bytes	Alto	El alto en píxeles de la imagen.
2 bytes	Planos	Los planos del dispositivo de salida.
2 bytes	Profundidad	Indica la profundidad del color en bits por píxel.
4 bytes	Compresión	Tipo de compresión. En la práctica no se encuentran BMPs comprimidos.
4 bytes	Tamaño de la imagen	Tamaño en bytes de la imagen (paleta y píxeles)
4 bytes	Píxeles por metro (h)	Píxeles por metro en horizontal
4 bytes	Píxeles por metro (v)	Píxeles por metro en vertical
4 bytes	Colores usados	Cantidad de colores usados por la imagen. Si el valor es 0, el programa lo calcula a partir de la profundidad.
4 bytes	Colores Importantes	Cantidad de colores importantes usados por la imagen. Si es 0 toma todos los colores como importantes.

2.4.- Paleta de Colores.

A continuación de la cabecera de información nos encontraríamos con la paleta de colores. En imágenes de 8 bits se disponen de un total de 256 colores que están incluidos en la paleta de colores, que identifica que color corresponde a un número. Por el contrario, en imágenes de 24 bits, se disponen de 3 bytes correspondientes a los valores RGB de la imagen. El programa, detecta si la imagen es de 8 bits y carga la paleta de colores para poder representar la imagen. En caso contrario, el color se obtiene a partir de las componentes RGB.

2.5.- Datos de imagen.

En imágenes de 24 bits, las componentes RGB se guardan en orden inverso. La siguiente tabla muestra la estructura de un píxel de la imagen:

1 byte	1 byte	1 byte
Color Azul (B)	Color Verde (G)	Color Rojo (R)

3.- Introducción del texto dentro de la imagen BMP

3.1.- Codificación de los caracteres en los píxeles de la imagen.

Como se ha indicado en el apartado 2.5, cada píxel esta formado por un conjunto de 3 bytes correspondientes a las tres componentes cromáticas.

Para introducir un carácter dentro de los píxeles de la imagen, seria necesario modificar esos 3 bytes de tal modo que contengan la información del carácter. Para que los cambios sean inapreciables al ojo humano, solo se cambiará el bit menos significativo. Se puede cambiar con una relación 3.1, 3.2 o 3.3, es decir, cambiando solo el bit menos significativo de un solo byte, de dos bytes o de los 3 bytes. En el presente proyecto se ha decidido hacerlo con una relación 3.3, de tal modo que en una misma imagen se pueden introducir 3 veces mas datos, y se ha comprobado que aun con una relación 3.3 los cambios en el color de la imagen no se aprecian al ojo humano.

A continuación se muestra una tabla con la codificación 3.3 de un carácter en una imagen blanca:

Píxel	Color	Bits Píxel	Bits Carácter (A)	Bits Píxel Final	Bits Carácter	
1	B	1111 1111	0100 000 1	1111 111 1	1	A
	G	1111 1111	0100 000 0 1	1111 111 0	0	
	R	1111 1111	0100 0 001	1111 111 0	0	
2	B	1111 1111	0100 0 001	1111 111 0	0	
	G	1111 1111	01 0 0001	1111 111 0	0	
	R	1111 1111	0 1 00 0001	1111 111 0	0	
3	B	1111 1111	0 100 0001	1111 111 1	1	
	G	1111 1111	0 100 0001	1111 111 0	0	
	R	1111 1111	XXXX XXXX	1111 111 X	X	

(*): Las X mostradas en la tabla anterior pertenecen a los bits del siguiente carácter del texto. En el ejemplo nos hemos limitado a codificar el carácter A.

Como se puede apreciar, solo cambia el bit menos significativo, de tal modo, que hay ciertos colores que permanecen exactamente igual que en la imagen original, y otros que cambian el color por el color anterior o siguiente.

Se puede apreciar, por ejemplo, que en el píxel número 1, el color azul permanece intacto, mientras que los otros dos cambian, de tal modo que el color rojo y verde pasan de ser el color 255 a ser el 254. Un cambio totalmente inapreciable al ojo humano.

4.- Software

4.1.- Lenguaje de Programación empleado.

En vista de que actualmente todas las aplicaciones informáticas optan por una interfaz gráfica de mayor o menos complejidad, en el presente proyecto se ha optado por un lenguaje que permita esta interfaz gráfica.

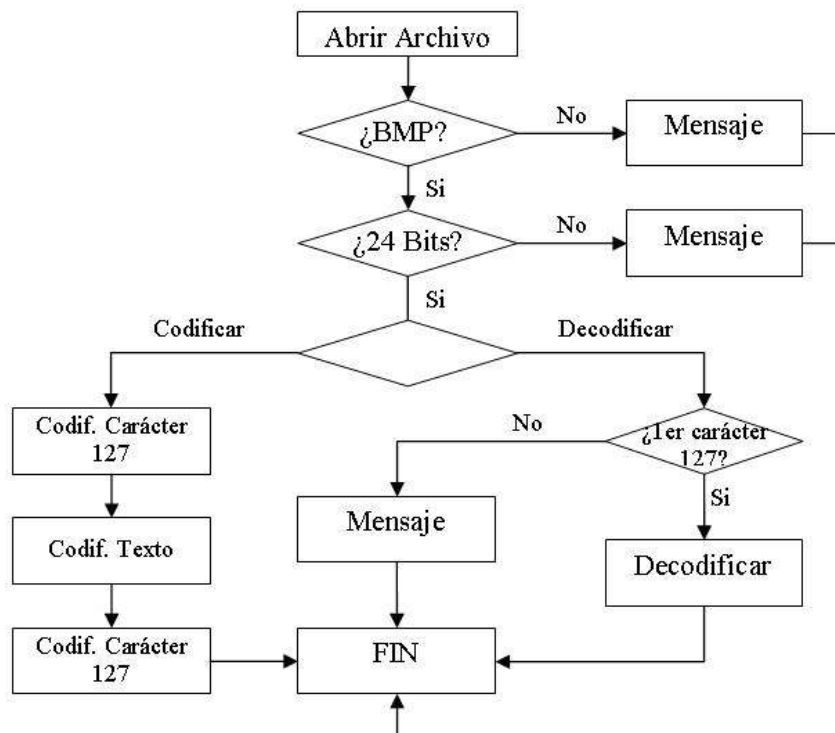
Los dos lenguajes mas empleados en ese caso son Visual Basic y Visual C++. En el presente proyecto se ha optado por Visual Basic, de la compañía Microsoft, puesto que se trata de un lenguaje de alto nivel, a diferencia de Visual C++, con la ventaja de que a nivel de programación es mucho mas sencillo y para el software que se desea realizar se poseen las funciones apropiadas para la realización del objetivo del software.

El mayor inconveniente que posee Visual Basic es la necesidad de librerías externas (*.dll, *.ocx,...) que necesita para poder realizar todas sus funciones correctamente.

Este inconveniente se resuelve utilizando el generador de instaladores InstallShield Express 5.0 que permite abrir un proyecto de Visual Basic y agregar al instalador todas las librerías necesarias para el programa, con lo cual nos aseguramos que las librerías necesarias siempre estarán en el equipo informático tras la instalación del software.

4.2.- Principios de codificación del texto.

Para codificar el texto de la forma explicada en el apartado 3.1, el software deberá realizar las operaciones que se detallan en el siguiente diagrama:



Como puede apreciarse en el diagrama anterior, el software realiza la apertura en modo lectura del archivo imagen sobre el que se va a codificar el texto, para proceder a la

comprobación de la imagen. Si se trata de una imagen BMP real y si la profundidad de color es de 24 bits. Si se cumplen estos requisitos existen dos opciones:

- Codificar: El programa realiza la codificación del texto entre una cabecera de inicio y fin de texto. Para dicha cabecera se ha utilizado el carácter 127 (también codificado). La finalidad de dicha cabecera es la identificación de texto codificado sobre la imagen y la identificación del final del texto codificado, para evitar la obtención de caracteres aleatorios producidos por los píxeles no codificados de la imagen.
- Decodificar: Identifica si el primer carácter es el 127, por lo tanto la imagen posee texto codificado, y si es así comienza la decodificación de la imagen.

Estos dos procesos de codificación y decodificación quedan explicados en profundidad en los apartados 4.4.2 y 4.4.3

4.3.- Obtención de los bytes de la imagen.

4.3.1.- Funciones Get & Put

Para obtener los bytes de la imagen para su manejo, Visual Basic dispone de 2 funciones que permiten la apertura de un archivo en modo binario, y obtener o introducir en el bytes. Visual Basic no puede manejar directamente bits, pero en el apartado 4.5 se explica como se puede hacer.

Para obtener los bytes de un archivo, es necesario abrir dicho archivo en modo binario, y posteriormente obtener o introducir bytes. A continuación se muestra un código de ejemplo:

```
Open "archivo" For Binary Access Read as #1
Datos = Space(10)
Get #1, ,Datos
Open "archivo2" For Binary Access Write as #2
Put #2,LOF(2) ,Datos
Close #1
Close #2
```

En el anterior ejemplo, se ha abierto un fichero llamado archivo en modo lectura. El #1 se trata del identificador de dicho archivo, es decir, dicho archivo se reconocerá como el archivo 1. Posteriormente se ha declarado la variable Datos con una longitud de 10 bytes (Space(10)). Con la función **Get**, se toman 10 bytes de "archivo". El espacio central entre las dos comillas es el lugar donde introducir a partir de que byte obtener, como se ha dejado en blanco ha comenzado desde el ultimo registro que se obtuvo. Como se acaba de abrir el archivo, ha comenzado desde el registro 0.

El paso siguiente es abrir el fichero "archivo2" en modo de escritura y se le asigna el identificador 2. A continuación, mediante la función **Put**, se introduce el valor de la variable Datos en el fichero "archivo2". Dichos datos se introducen al final del archivo, puesto que entre las comillas hemos puesto la función "**LOF**" que equivale al tamaño total de un archivo abierto, en este caso el 2, por lo tanto, los datos los introduce a partir del ultimo registro del archivo.

Para finalizar, cierra los dos archivos y termina el proceso.

En el caso que el archivo no se cierre el proceso seguirá obteniendo bytes a partir del ultimo obtenido, por lo tanto, si se vuelve a usar una función Get a continuación, se obtendrán los bytes a partir del 11 incluido.

4.4. Codificación del texto en la imagen.

4.4.1.- Funciones lógicas And y Or a nivel de bit

Uno de los inconvenientes que presenta Visual Basic respecto a lenguajes de bajo nivel es la incapacidad de manejar bits de forma directa. Este inconveniente puede ser resultado con las funciones lógicas And y Or, ya que actúan a nivel de bit. Con estas funciones se puede cambiar u obtener cualquier bit de cualquier byte. A continuación se detallan los modos de utilización de ambas funciones para todos los casos.

Escribir bits:

- Poner un bit a uno:

Para poner un bit a uno se aplicará la función Or del modo que se detalla a continuación:

Ej.:

1101 1100	1000 0111	0011 0001
Or	Or	Or
0000 0001	0010 0000	1000 0000
-----	-----	-----
1101 1101	1010 0111	1011 0001

Como puede apreciarse, y según la tabla de verdad de la operación Or (fig.1) para cambiar un bit a uno se debe realizar la operación Or del siguiente modo:

- Cambiar bit 1: byte Or **0000 0001**
- Cambiar bit 2: byte Or **0000 0010**
- Cambiar bit 3: byte Or **0000 0100**
- Cambiar bit 4: byte Or **0000 1000**
- Cambiar bit 5: byte Or **0001 0000**
- Cambiar bit 6: byte Or **0010 0000**
- Cambiar bit 7: byte Or **0100 0000**
- Cambiar bit 8: byte Or **1000 0000**

B	A	S
0	0	0
0	1	1
1	0	1
1	1	1

Fig. 1
Tabla de verdad de la función Or

- Poner un bit a cero:

Para poner un bit a cero, se aplicará la función And del modo que se detalla a continuación:

Ej.:

1011 001 1	0111 1 000	0 1 11 1100
And	And	And
1111 1110	1111 0111	1011 1111
-----	-----	-----
1011 001 0	0111 0 000	0 011 1100

Como puede apreciarse, y según la tabla de verdad de la función And (fig. 2) para cambiar un bit a cero se debe realizar la operación And de los siguientes modos:

- Cambiar bit 1: byte And **1111 1110**
- Cambiar bit 2: byte And **1111 1101**
- Cambiar bit 3: byte And **1111 1011**
- Cambiar bit 4: byte And **1111 0111**
- Cambiar bit 5: byte And **1110 1111**
- Cambiar bit 6: byte And **1101 1111**
- Cambiar bit 7: byte And **1011 1111**
- Cambiar bit 8: byte And **0111 1111**

B	A	S
0	0	0
0	1	0
1	0	0
1	1	1

Fig. 2
Tabla de verdad de la función And

- Obtener un bit de un determinado byte:

Para obtener el valor de un determinado bit, se aplica la función And del modo detallado a continuación:

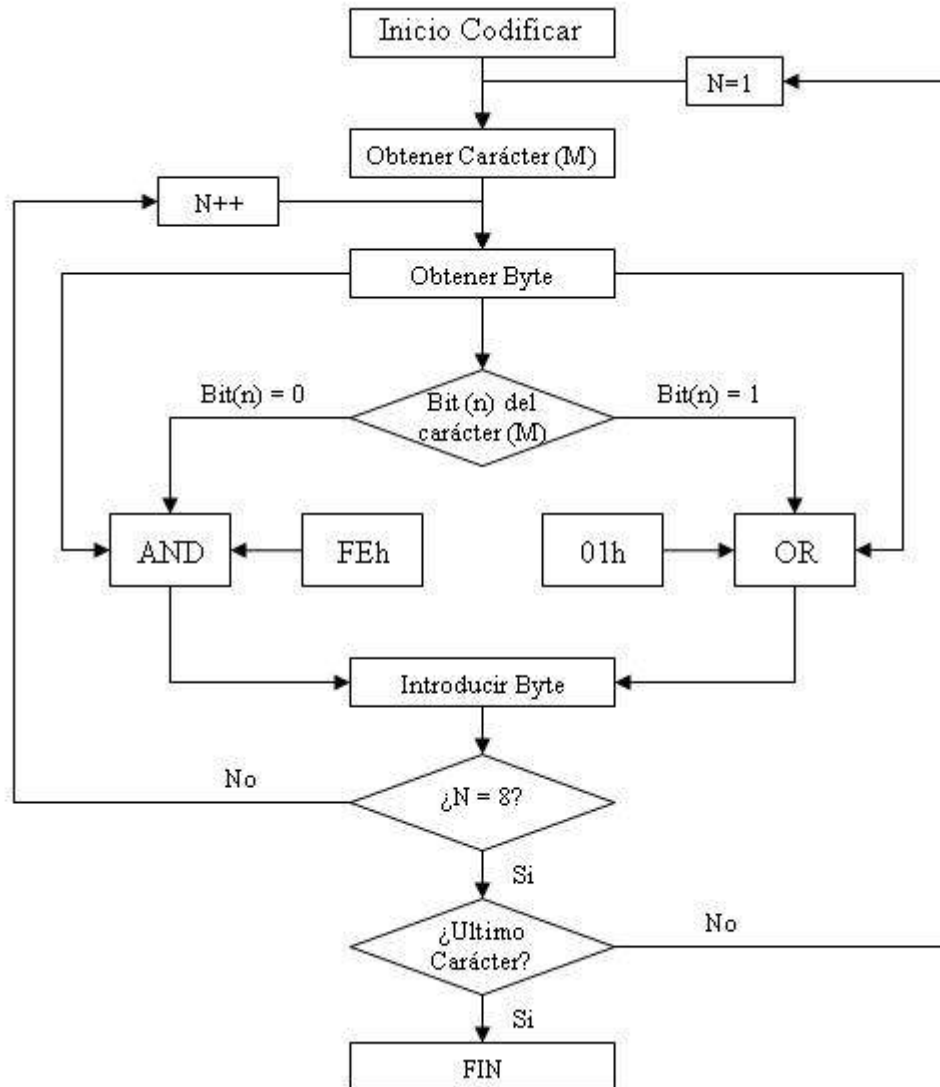
Ej.:

1111 011 0	1001 1 011	01 0 1 1111
And	And	And
0000 0001	0000 1000	0010 0000
-----	-----	-----
0000 000 0	0000 1 000	00 0 0 0000

Como se aprecia el ejemplo anterior, en el caso en que el bit sea 0, el resultado de la operación And será cero. En caso de que sea 1, el resultado de la operación And será distinto de 0. De este modo se puede saber en cualquier byte el estado de sus bits.

4.4.2.- Codificación de caracteres en la imagen.

Basándose en lo expuesto en el apartado 4.4.1, el software del presente proyecto realiza las siguientes acciones para la codificación del texto:



El software recorre todos los caracteres del texto (Carácter(M)), aplicándole a cada uno la función AND para obtener el estado de cada bit, y a partir de ahí aplicar la función OR o AND al byte correspondiente a la imagen, de tal modo que el texto quede codificado del modo explicado en el apartado 3.1.

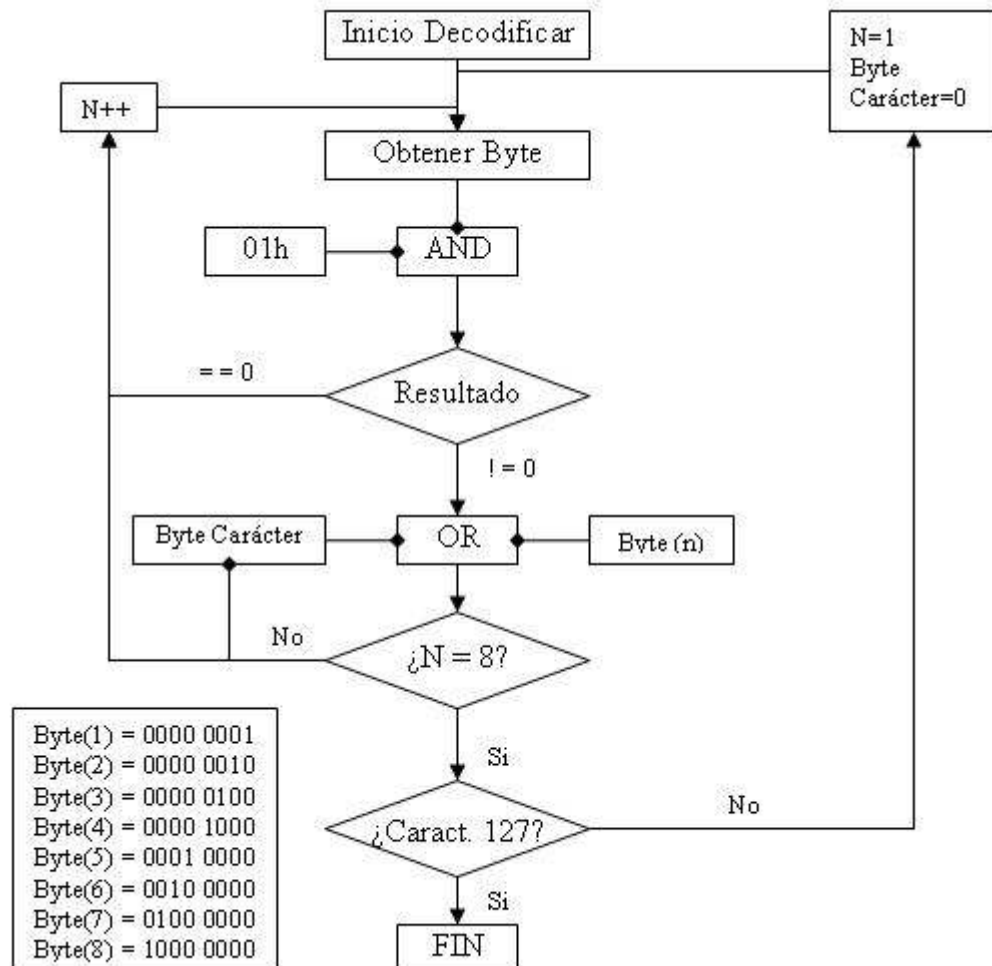
En el caso en que se hayan codificado todos los bits de la imagen (N=8), y tras comprobar que el carácter codificado no se trate del ultimo carácter del texto, se pasa a obtener el siguiente carácter y se comienza un nuevo bucle para volver a codificar el carácter.

En el caso en que no se haya codificado completamente el carácter, se toma el valor de su siguiente byte, se toma otro nuevo byte de la imagen, y se le aplica la función AND u OR dependiendo del estado de dicho carácter.

Todo esto puede ser estudiado con mas profundidad en el apartado 6.1 donde se detalla completamente el código del presente proyecto.

4.4.3.- Decodificación de caracteres en la imagen.

Basándose en lo expuesto en el apartado 4.4.1, el software del presente proyecto realiza las siguientes acciones para la codificación del texto:



Como puede observarse en el diagrama anterior, en el momento de decodificar, el software obtiene un byte de la imagen, al cual se le aplica la función AND con 01h, de tal modo que obtenemos el valor del ultimo bit, correspondiente a una parte del carácter. En el caso de que el resultado sea igual a uno, se aplica la función OR a una variable (que en principio es igual a 0) con una constante (mostradas en la tabla) de tal modo que se va reconstruyendo el byte perteneciente al carácter.

En el caso en que el resultado sea igual a 0, se pasará directamente a obtener el siguiente byte, ya que al ser cero, no hay que aplicar ninguna operación a la variable, puesto que la función OR introduce los unos necesarios, no los ceros.

En el caso de que se haya completado un carácter (N=8) comprobará si el carácter decodificado es el 127, usado como cabecera de fin de texto. De ser así el programa finalizará su función. En caso contrario reseteará todas las variables y volverá a decodificar el siguiente carácter.

5.- Guía de Usuario.

5.1.- Requisitos Mínimos

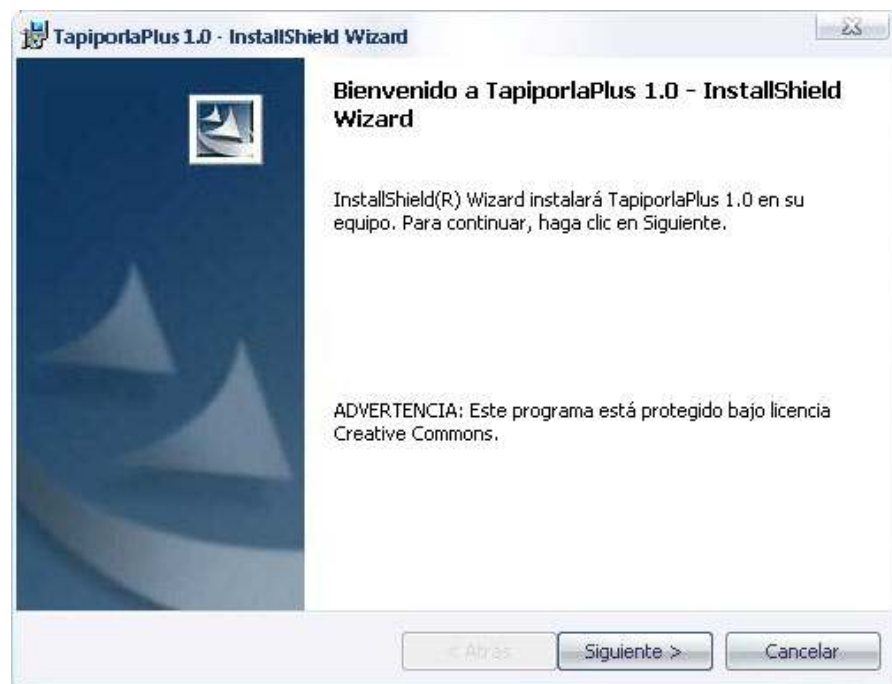
- Procesador Intel / AMD 350 Mhz o superior.
- 32 Mb de RAM.
- Sistema Operativo Win98 SE, WinMe, Win2000 o WinXP.
- 20 Mb de espacio libre en disco

5.2.- Instalación.

Para proceder a la instalación de TapiportaPlus v.1.0 deberá de introducir el cd-rom en la unidad correspondiente.

A continuación le aparecerá una pantalla (fig. 1) donde deberá hacer clic sobre “Instalar”.

Tras esperar algunos segundos, le aparecerá en pantalla una ventana como la siguiente:



Deberá hacer clic sobre siguiente para pasar a la siguiente pantalla de la instalación:

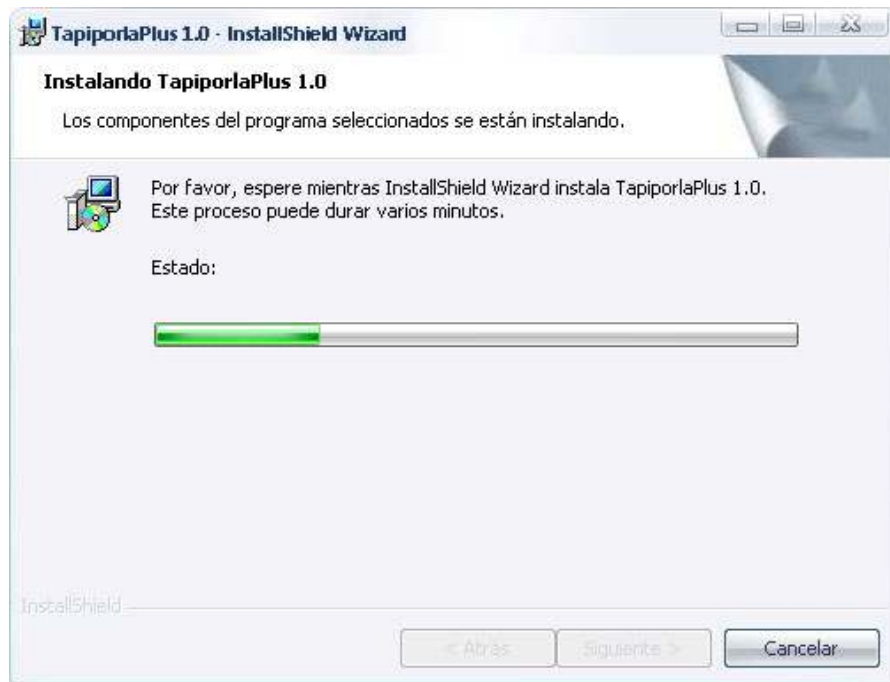


Después de la lectura de la licencia de Creative Commons deberá aceptar sus condiciones si desea proceder con la instalación de TapiporlaPlus.

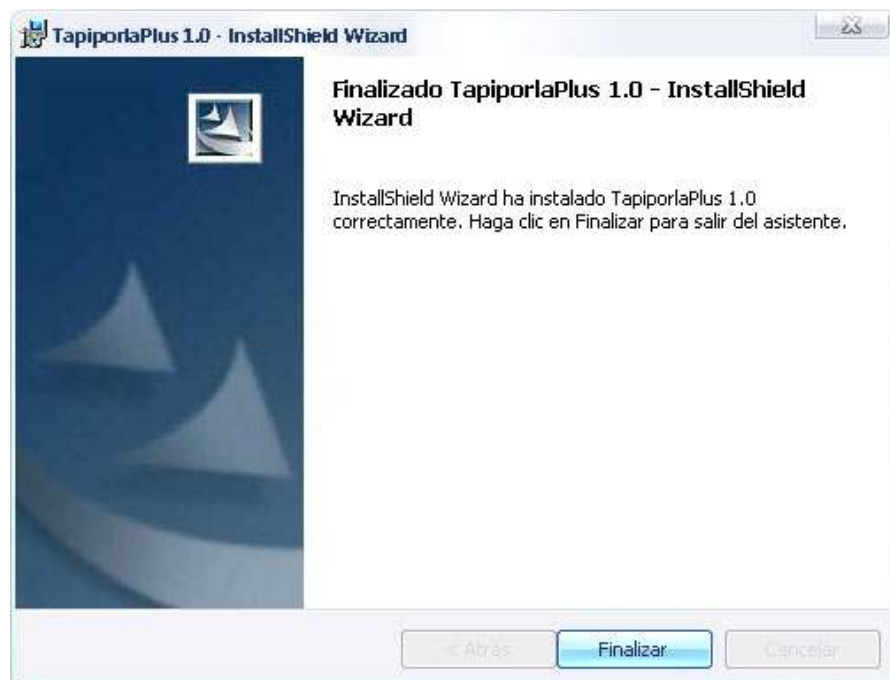
Una vez aceptadas las condiciones, podrá hacer clic sobre siguiente, lo cual le llevará a la siguiente ventana:



En esta ventana se le informará sobre la configuración de la instalación. Para procesar la instalación deberá hacer clic sobre "Instalar", lo cual comenzará la instalación del software como puede apreciarse en la siguiente imagen:



Deberá permanecer a la espera mientras se completa el proceso de instalación. Una vez finalizado solo le restará hacer clic sobre “finalizar” en la siguiente pantalla.



De este modo habrá sido instalado TapiporlaPlus v.1.0 en su ordenador. Encontrará los accesos directos al programa en el escritorio así como en: Inicio/Programas/TapiporlaPlus

5.3.- Entorno.

Tras la instalación de TapiportaPlus, usted ya puede ejecutar el programa. Una vez ejecutado aparecerá en pantalla la ventana correspondiente al menú principal del programa, el cual es el siguiente:



Como puede apreciarse, existen 3 botones, correspondientes a las siguientes acciones:



Este botón nos llevará al menú de Encriptar, el cual nos introducirá un texto dentro de una imagen BMP.



Este botón nos llevará al menú de Desencriptar, el cual nos obtendrá un texto de una imagen BMP previamente encriptada.



Este botón nos mostrará el documento de Ayuda del programa, donde se pueden encontrar estas mismas palabras.

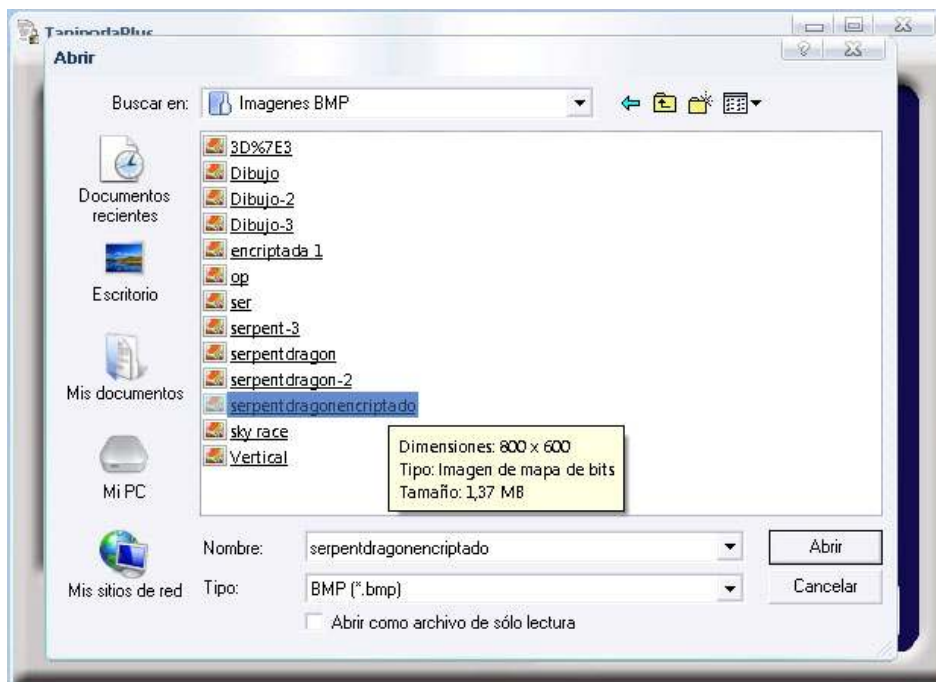
5.4.- Encriptar.

Para proceder a la encriptación de un texto sobre una imagen se deben seguir los siguientes pasos:

Una vez hecho clic sobre el botón de encriptar (apartado 5.3) aparecerá la siguiente pantalla:



Como se especifica arriba, debe seleccionar una imagen sobre la que introducir el texto, para ello deberá hacer clic sobre el icono de la carpeta y buscar la imagen deseada.



En cualquier momento el proceso puede ser abortado pulsando sobre el icono de Stop.

Una vez seleccionada la imagen, aparecerá la siguiente pantalla:



Como puede apreciarse, la imagen deseada aparece en el programa. Recuerde que esta imagen puede ser cambiada haciendo clic de nuevo sobre el icono de la carpeta.

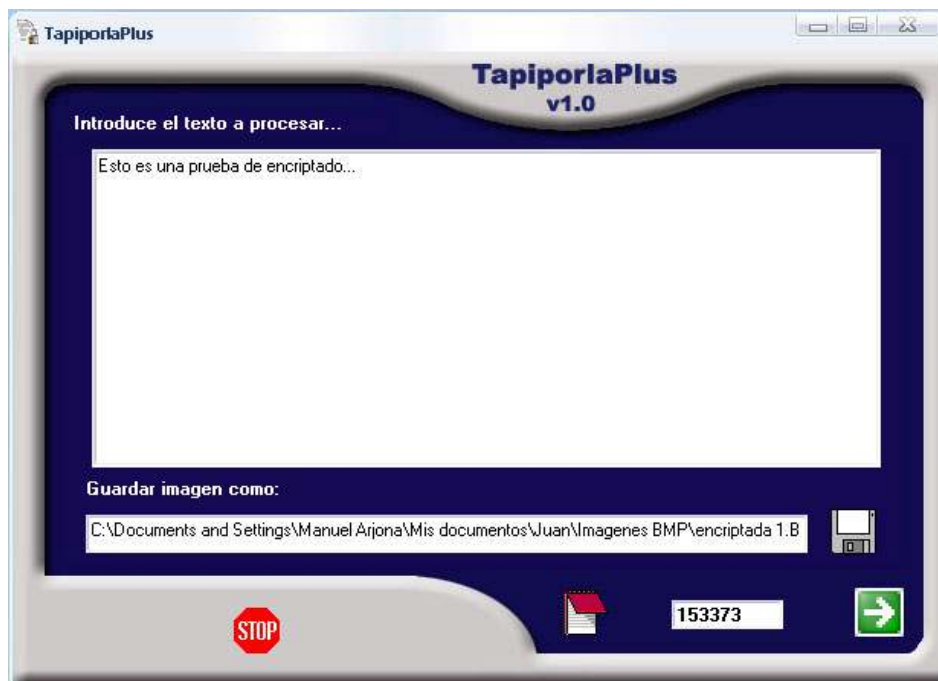
Para continuar con el proceso se debe hacer clic sobre el botón de la flecha, lo cual nos llevará a la siguiente pantalla:



En el primer cuadro de texto se debe introducir el texto que deseamos encriptar en la imagen anteriormente seleccionada. Este texto puede ser cargado desde un archivo *.txt haciendo clic sobre el botón de la libreta roja y seleccionando el archivo *.txt deseado.

A continuación debe especificarse la imagen de destino. Dicha imagen será la que posea el texto en su interior. Si no se especifica ninguna imagen, el programa automáticamente guardará la imagen en: "C: \prueba.bmp".

En esta pantalla aparece también un contador, el cual nos indica la cantidad de caracteres que pueden ser introducidos en la imagen. Dicho contador irá decreciendo a medida que se vayan introduciendo caracteres.



Para continuar con el proceso debe hacer clic sobre el botón de la flecha, lo cual nos llevará a la siguiente pantalla:



Como puede apreciar, el programa le va informando sobre todo el proceso de encriptación. En este caso todo el proceso ha sido satisfactorio. De no ser así, el propio programa le informará con un mensaje de error.

Tras finalizar el proceso de forma satisfactorio, sonará una melodía por los altavoces que le indicará que todo ha sido correcto, y tras pulsar sobre aceptar, el programa volverá al menú principal.

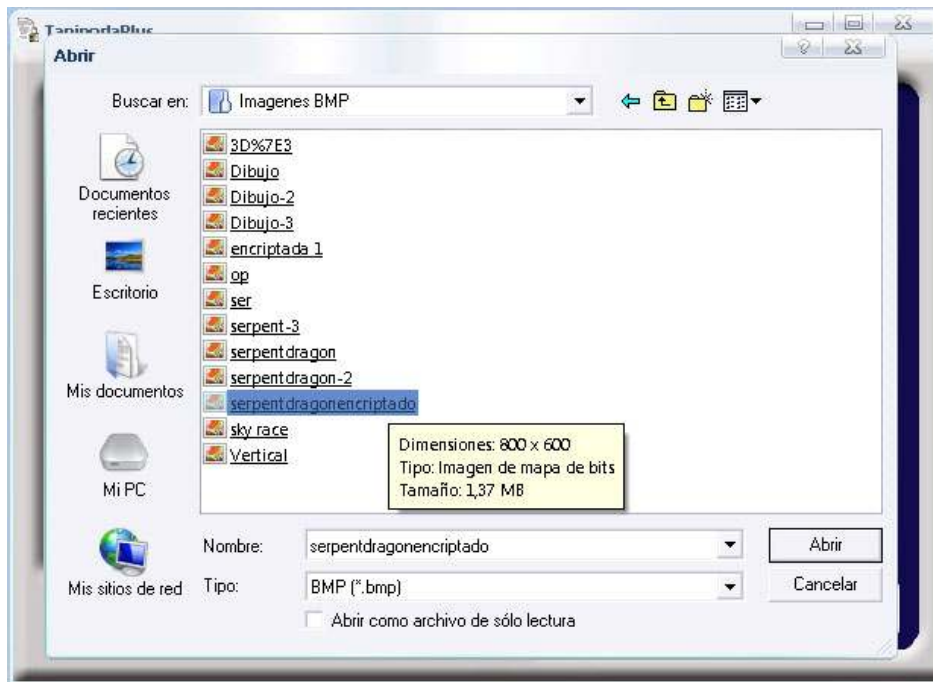
5.5.- Desencriptar.

Para proceder a la desencriptación de un texto sobre una imagen se deben seguir los siguientes pasos:

Una vez hecho clic sobre el botón de desencriptar (apartado 5.3) aparecerá la siguiente pantalla:



Como se especifica arriba, debe seleccionar una imagen sobre la que obtener el texto, para ello deberá hacer clic sobre el icono de la carpeta y buscar la imagen deseada.



En cualquier momento el proceso puede ser abortado pulsando sobre el icono de Stop.

Una vez seleccionada la imagen, aparecerá la siguiente pantalla:



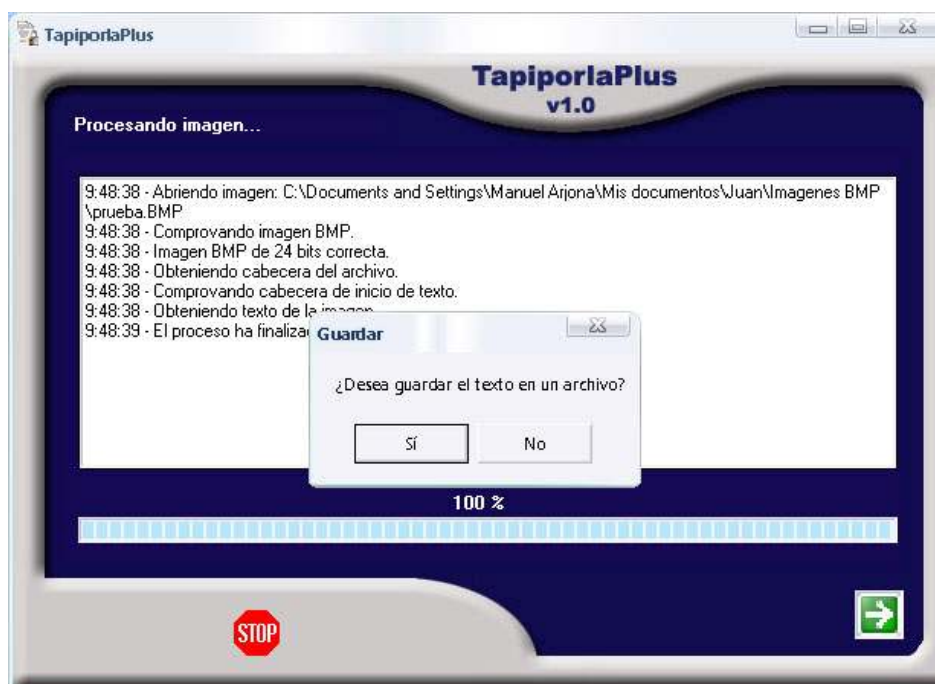
Como puede apreciarse, la imagen deseada aparece en el programa. Recuerde que esta imagen puede ser cambiada haciendo clic de nuevo sobre el icono de la carpeta.

Para continuar con el proceso se debe hacer clic sobre el botón de la flecha, lo cual nos llevará a la siguiente pantalla:



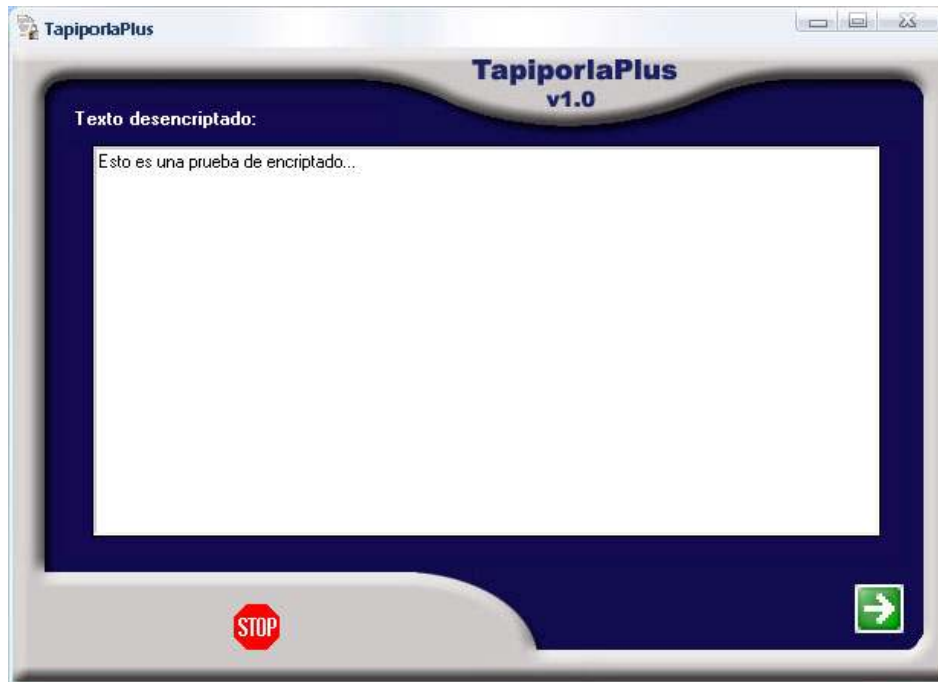
Tras permanecer a la espera de que finalice todo el proceso, aparecerá un mensaje y una melodía notificando que el proceso ha sido satisfactorio. En caso contrario, y al igual que el proceso de encriptación, el programa notificará con un mensaje el error producido.

Tras pulsar sobre aceptar aparecerá un nuevo mensaje como puede apreciarse a continuación:



Como puede apreciarse, el programa nos pregunta sobre si guardar el texto sobre un archivo *.txt. En caso afirmativo, nos pedirá la ruta donde guardarlo y tras seleccionarla, nos creará un archivo con el texto obtenido de la imagen.

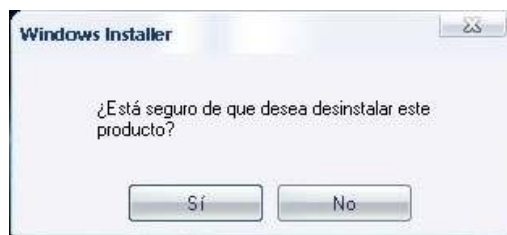
Realizadas estas operaciones pulse siguiente para visualizar el texto obtenido de la imagen, tal como se aprecia en la siguiente imagen:



Para finalizar, pulse sobre siguiente y el programa volverá al menú de inicio, donde podrá volver a realizar las operaciones de encriptado/descriptado.

5.6.- Desinstalación.

Si por cualquier motivo usted desea desinstalar TapiporlaPlus de su equipo, solo bastará con dirigirse a: Inicio/Programas/TapiporlaPlus 1.0/Uninstall TapiporlaPlus 1.0 y seguir los siguientes pasos:



Si usted esta realmente seguro de querer desinstalar TapiporlaPlus de su equipo, pulse sobre “Si”, en caso contrario pulse sobre “No”.

Si usted ha pulsado sobre “Si” el programa se desinstalará automáticamente al cabo de unos segundos que dura el proceso.

6.- Anexos.

6.1.- Código completo de la aplicación.

```
Private Sub Form_Load()  
'Ajustando el tamaño del formulario  
TapiporlaPlus.Height = 6720  
TapiporlaPlus.Width = 9270  
End Sub
```

```
Private Sub Form_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)  
'Eliminando bordes de los botones  
Image1.BorderStyle = 0  
Image2.BorderStyle = 0  
Image3.BorderStyle = 0  
Image6.BorderStyle = 0  
Image8.BorderStyle = 0  
Image9.BorderStyle = 0  
Image10.BorderStyle = 0  
Image11.BorderStyle = 0  
Image12.BorderStyle = 0  
End Sub
```

```
Private Sub Image1_Click()  
'Ejecutar archivo de ayuda  
ayuda = App.Path & "\Ayuda\Ayuda.chm"  
a = Shell("hh.exe " & ayuda, vbNormalFocus)  
End Sub
```

```
Private Sub Image12_Click()  
'Volver al menú principal  
msg = MsgBox("¿Desea volver al menú principal?", vbYesNo, "TapiporlaPlus")  
If msg = 6 Then  
Call borrar  
End If  
End Sub
```

```
Private Sub Image2_Click()  
'Mostrando el menú de encriptar  
Call borrar  
agregar.Left = 480  
agregar.Top = 960  
agregar.Visible = True  
Label1.Visible = True  
Label1.Caption = "Introduce la imagen a encriptar..."  
Image6.Visible = True  
Image8.Visible = True  
Image12.Visible = True  
Image9.Visible = False  
Image2.Visible = False  
Image3.Visible = False  
End Sub
```

```
Private Sub Image3_Click()  
'Mostrando el menú de desencriptar  
Call borrar  
agregar.Top = 960  
agregar.Left = 480  
agregar.Visible = True
```

```
Label1.Visible = True
Label1.Caption = "Introduce la imagen a desencriptar..."
Image12.Visible = True
Image6.Visible = True
Image8.Visible = False
Image9.Visible = True
Image2.Visible = False
Image3.Visible = False
End Sub
```

'Poniendo borde al boton sobre el cual se situa el cursor

```
Private Sub Image12_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
Image12.BorderStyle = 1
End Sub
Private Sub Image10_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
Image10.BorderStyle = 1
End Sub
Private Sub Image11_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
Image11.BorderStyle = 1
End Sub
Private Sub Image8_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
Image8.BorderStyle = 1
End Sub
Private Sub Image9_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
Image9.BorderStyle = 1
End Sub
Private Sub Image3_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
Image3.BorderStyle = 1
End Sub
Private Sub Image2_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
Image2.BorderStyle = 1
End Sub
Private Sub Image1_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
Image1.BorderStyle = 1
End Sub
Private Sub Image6_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
Image6.BorderStyle = 1
End Sub
```

```
Private Sub Image11_Click()
'Buscando un archivo de texto
On Error Resume Next
Set fso = CreateObject("Scripting.FileSystemObject")
Dim im
abrir.FileName = ""
abrir.Filter = "TXT (*.txt)|*.TXT"
abrir.ShowOpen
With abrir
im = .FileName
End With
Set f = fso.OpenTextFile(im, 1) 'Abriendo el archivo en modo lectura
s = f.ReadAll 'Leyendo todo el contenido del archivo
Text1.Text = s 'Cargando el contenido en el cuadro de texto
End Sub
```

```
Private Sub Image10_Click()
'Estableciendo imagen de destino.
abrir.FileName = ""
Dim im
abrir.Filter = "BMP (*.bmp)|*.BMP"
```

```
abrir.ShowSave
With abrir
im = .FileName
End With
Text3.Text = im
End Sub

Private Sub Image6_Click()
'Estableciendo dimensiones iniciales
foto.Left = 420
foto.Height = 3855
foto.Width = 7575
Image7.Height = 253
Image7.Width = 501
Dim im
abrir.FileName = ""
abrir.Filter = "BMP (*.bmp)|*.BMP"
abrir.ShowOpen
With abrir
im = .FileName
End With
'Onteniendo dimensiones de la imagen
Image7.ToolTipText = im
Set objpic = LoadPicture(im)
a = objpic.Width
b = objpic.Height
If b > a Then
'Ajustando imagen en el caso que sea vertical
d = Image7.Height
c = a / b
MsgBox c
Image7.Height = d * c
foto.Height = foto.Height * c
h = foto.Height - 15
tt = (3855 - h) / 2
foto.Top = foto.Top + tt - (240 / 2)
End If
If a > b Then
'Ajustando imagen en el caso que sea horizontal
d = Image7.Width
c = b / a
Image7.Width = d * c
foto.Width = foto.Width * c
w = foto.Width - 15
ll = (7575 - w) / 2
foto.Left = foto.Left + ll - (120 / 2)
End If
Image7.Stretch = True
Set Image7.Picture = objpic 'Cargando imagen en el cuadro de imagen
foto.Visible = True
'Calculando total de caracteres que pueden ser introducidos en la imagen
alto = Round((objpic.Height / 26) * 0.982)
ancho = Round((objpic.Width / 26) * 0.982)
total = Round((alto * ancho) / 3)
Text1.MaxLength = total
Text4.Text = total
End Sub
```

```
Private Sub Image8_Click()  
'MENU ENCRYPTAR  
'Si no se selecciona ninguna imagen...  
If Image7.Picture = 0 Then  
MsgBox "Debes seleccionar una imagen", vbInformation, "TapiporlaPlus"  
Exit Sub  
End If  
Image1.Visible = False  
Image2.Visible = False  
Image3.Visible = False  
If foto.Visible = True Then  
  'Mostrando ventana para introducir el texto y la ruta de destino  
  foto.Visible = False  
  Text1.Visible = True  
  Text3.Visible = True  
  Text4.Visible = True  
  Label2.Visible = True  
  Image10.Visible = True  
  Image6.Visible = False  
  Image11.Visible = True  
  Label1.Caption = "Introduce el texto a procesar..."  
  Exit Sub  
End If  
'Si no se ha introducido ningun texto...  
If Text1.Text = "" Then  
MsgBox "Debes introducir un texto en el cuadro", vbInformation, "TapiporlaPlus"  
Exit Sub  
End If  
If Text1.Visible = True Then  
  'Mostrando ventana de proceso de encriptación  
  Text1.Visible = False  
  Text3.Visible = False  
  Text4.Visible = False  
  Image10.Visible = False  
  Image11.Visible = False  
  Label2.Visible = False  
  Text2.Visible = True  
  Progress1.Visible = True  
  Label1.Caption = "Procesando imagen..."  
  Call encriptar  
  Image8.Visible = False  
  Exit Sub  
End If  
End Sub
```

```
Private Sub Image9_Click()  
'MENU DESENCRIPTAR  
'Si no existe una imagen para desencriptar...  
If Image7.Picture = 0 Then  
MsgBox "Debes seleccionar una imagen", vbInformation, "TapiporlaPlus"  
Exit Sub  
End If  
Image1.Visible = False  
Image2.Visible = False  
Image3.Visible = False  
If foto.Visible = True Then  
  'Mostrando ventana de proceso de desencriptación  
  foto.Visible = False  
  Text2.Visible = True  
  Progress1.Visible = True
```

```
Image6.Visible = False
Label1.Caption = "Procesando imagen..."
Call desencriptar
Exit Sub
End If
If Text2.Visible = True Then
    'Mostrando ventana de texto desencriptado
    Text2.Visible = False
    Text1.Visible = True
    Progress1.Visible = False
    Label3.Caption = ""
    Label1.Caption = "Texto desencriptado:"
    Text1.Height = 3855
    Exit Sub
End If
If Text1.Visible = True Then
Image9.Visible = False
Call borrar
End If
End Sub

Private Sub encriptar()
    'Declarando variables
    Image8.Visible = False
    Image12.Visible = False
    imagen = Image7.ToolTipText
    If Text3.Text = "" Then
        archivo = "C:\prueba.bmp"
    Else
        archivo = Text3.Text
    End If
    Dim datos As String
    Dim datos2 As String
    Dim bmp As String
    Dim bmp2 As Byte
    Dim byte1 As Byte
    Dim byte2 As Byte
    'Comprovando imagen bmp...
    Text2.Text = Hour(Now) & ":" & Minute(Now) & ":" & Second(Now) & " - Abriendo imagen: " &
    Image7.ToolTipText
    Open imagen For Binary Access Read As #1
    bmp = Space(2)
    Get #1, , bmp
    If Not bmp = "BM" Then
        MsgBox "La imagen introducida no se trata de una imagen BMP", vbCritical, "TapiporlaPlus"
        Text2.Text = Text2.Text & Chr(13) & Hour(Now) & ":" & Minute(Now) & ":" & Second(Now) & " - La
imagen introducida no se trata de un archivo BMP."
        Text2.Text = Text2.Text & Chr(13) & Hour(Now) & ":" & Minute(Now) & ":" & Second(Now) & " - El
proceso a finalizado inesperadamente."
        Call borrar
        Exit Sub
    End If
    'Comprovando imagen 24 bits...
    bmp = Space(26)
    Get #1, , bmp
    Get #1, , bmp2
    If Not bmp2 = 24 Then
        MsgBox "La imagen introducida no se trata de una imagen de 24 bits", vbCritical, "TapiporlaPlus"
        Text2.Text = Text2.Text & Chr(13) & Hour(Now) & ":" & Minute(Now) & ":" & Second(Now) & " - La
imagen introducida no se trata de un archivo BMP de 24 bits."
    End If
End Sub
```

Proyecto Encriptación de texto sobre imágenes BMP mediante software

```
Text2.Text = Text2.Text & Chr(13) & Hour(Now) & ":" & Minute(Now) & ":" & Second(Now) & " - El
proceso a finalizado inesperadamente."
Call borrar
Exit Sub
End If
Close #1
Text2.Text = Text2.Text & Chr(13) & Hour(Now) & ":" & Minute(Now) & ":" & Second(Now) & " - Imagen
BMP de 24 bits correcta."
Open imagen For Binary Access Read As #1
datos = Space(14) 'Variable que almacena la cabecera de la imagen
Text2.Text = Text2.Text & Chr(13) & Hour(Now) & ":" & Minute(Now) & ":" & Second(Now) & " -
Obteniendo cabecera del archivo."
Get #1, , datos 'Obteniendo bytes de la cabecera
Text2.Text = Text2.Text & Chr(13) & Hour(Now) & ":" & Minute(Now) & ":" & Second(Now) & " - Creando
archivo codificado a guardar"
Open archivo For Binary Access Write As #2
Put #2, , datos
Dim MisBits(8) As Byte, I As Integer
Progress1.Max = Len(Text1.Text)
Label3.Caption = "0 %"
Text2.Text = Text2.Text & Chr(13) & Hour(Now) & ":" & Minute(Now) & ":" & Second(Now) & " -
Introduciendo cabecera de inicio de texto."
'Obteniendo código binario del caracter de cabecera...
For I = 1 To 8
Get #1, , byte2 'Obteniendo bytes...
MisBits(I) = 178 And (2 ^ (I - 1))
If Not MisBits(I) = 0 Then
'Poniendo bit a uno...
byte2 = byte2 Or &H1
Else
'Poniendo bit a cero...
byte2 = byte2 And &HFE
End If
Put #2, , byte2 'Introduciendo byte en la imagen...
Next I
Text2.Text = Text2.Text & Chr(13) & Hour(Now) & ":" & Minute(Now) & ":" & Second(Now) & " -
Procesando píxeles de la imagen."
For J = 1 To Len(Text1.Text)
MiCaracter = Asc(Mid(Text1.Text, J, 1))
'Obteniendo los bytes del pixel
'Obteniendo código binario de cada caracter.
For I = 1 To 8
Get #1, , byte1 'Obteniendo bytes...
MisBits(I) = MiCaracter And (2 ^ (I - 1))
If Not MisBits(I) = 0 Then
'Poniendo bit a uno...
byte1 = byte1 Or &H1
Else
'Poniendo bit a cero...
byte1 = byte1 And &HFE
End If
Put #2, , byte1 'Introduciendo byte en la imagen...
Next I
DoEvents
Progress1.Value = J
Label3.Caption = "" & Round((Progress1.Value * 100) / Progress1.Max, 1) & " %"
Next J
Text2.Text = Text2.Text & Chr(13) & Hour(Now) & ":" & Minute(Now) & ":" & Second(Now) & " -
Introduciendo cabecera de fin de texto."
'Obteniendo código binario del caracter de cierre cabecera...
```

```
For I = 1 To 8
  Get #1, , byte2 'Obteniendo bytes...
  MisBits(I) = 178 And (2 ^ (I - 1))
  If Not MisBits(I) = 0 Then
    'Poniendo bit a uno...
    byte2 = byte2 Or &H1
  Else
    'Poniendo bit a cero...
    byte2 = byte2 And &HFE
  End If
  Put #2, , byte2 'Introduciendo byte en la imagen...
Next I
Text2.Text = Text2.Text & Chr(13) & Hour(Now) & ":" & Minute(Now) & ":" & Second(Now) & " -
Obteniendo final de archivo BMP"
datos2 = Space(LOF(1) - LOF(2)) 'Variable de tamaño igual al final del archivo
Get #1, , datos2 'Obteniendo final del archivo
Put #2, , datos2 'Introduciendo el final del archivo.
Text2.Text = Text2.Text & Chr(13) & Hour(Now) & ":" & Minute(Now) & ":" & Second(Now) & " -
Introduciendo final de archivo BMP"
Close #1 'Cerrando imagen a procesar
Close #2 'Cerrando imagen de destino
Text2.Text = Text2.Text & Chr(13) & Hour(Now) & ":" & Minute(Now) & ":" & Second(Now) & " -
Operacion realizada con éxito."
Label1.Caption = "Terminado con exito"
sonido.Command = "play" 'Reproducir sonido de finalizado
msg = MsgBox("La operación a terminado con éxito", vbInformation, "TapiporlaPlus")
If msg = 1 Then
  Call borrar 'Llamando a la funcion borrar
End If
Image8.Visible = True
End Sub

Private Sub desencriptar()
'Declarando variables y constantes
Text1.Text = ""
Image12.Visible = False
Image9.Visible = False
imagen = Image7.ToolTipText
archivo = "C:\prueba.bmp"
Dim datos As String
Dim datos2 As String
Dim bmp As String
Dim bmp2 As Byte
Dim byte1 As Byte
Dim byte2 As Byte
Dim MisBits(8) As Byte, bit(8) As Byte, I As Integer
bit(1) = &H1
bit(2) = &H2
bit(3) = &H4
bit(4) = &H8
bit(5) = &H10
bit(6) = &H20
bit(7) = &H40
bit(8) = &H80
Text2.Text = Hour(Now) & ":" & Minute(Now) & ":" & Second(Now) & " - Abriendo imagen: " &
Image7.ToolTipText
'Comprovando imagen bmp...
Open imagen For Binary Access Read As #1
Progress1.Max = LOF(1)
Label3.Caption = "0 %"
```

```
Text2.Text = Text2.Text & Chr(13) & Hour(Now) & ":" & Minute(Now) & ":" & Second(Now) & " -  
Comprovando imagen BMP."  
bmp = Space(2)  
Get #1, , bmp  
If Not bmp = "BM" Then  
    MsgBox "La imagen introducida no se trata de una imagen BMP", vbCritical, "TapiporlaPlus"  
    Text2.Text = Text2.Text & Chr(13) & Hour(Now) & ":" & Minute(Now) & ":" & Second(Now) & " - La  
imagen seleccionada no es correcta."  
    Text2.Text = Text2.Text & Chr(13) & Hour(Now) & ":" & Minute(Now) & ":" & Second(Now) & " - El  
proceso a finalizado inesperadamente."  
    Call borrar  
    Exit Sub  
End If  
  
'Comprovando imagen 24 bits...  
bmp = Space(26)  
Get #1, , bmp  
Get #1, , bmp2  
If Not bmp2 = 24 Then  
    MsgBox "La imagen introducida no se trata de una imagen de 24 bits", vbCritical, "TapiporlaPlus"  
    Text2.Text = Text2.Text & Chr(13) & Hour(Now) & ":" & Minute(Now) & ":" & Second(Now) & " - La  
imagen introducida no se trata de un archivo BMP de 24 bits."  
    Text2.Text = Text2.Text & Chr(13) & Hour(Now) & ":" & Minute(Now) & ":" & Second(Now) & " - El  
proceso a finalizado inesperadamente."  
    Call borrar  
    Exit Sub  
End If  
Close #1  
Text2.Text = Text2.Text & Chr(13) & Hour(Now) & ":" & Minute(Now) & ":" & Second(Now) & " - Imagen  
BMP de 24 bits correcta."  
Text2.Text = Text2.Text & Chr(13) & Hour(Now) & ":" & Minute(Now) & ":" & Second(Now) & " -  
Obteniendo cabecera del archivo."  
Open imagen For Binary Access Read As #1 'Abriendo imagen en modo lectura  
datos = Space(54) 'Variable que contendrá la cabecera  
Get #1, , datos 'Obteniendo cabecera del archivo  
Progress1.Value = 54  
Label3.Caption = "" & Round((Progress1.Value * 100) / Progress1.Max, 1) & " %" 'Calculando porcentaje de  
proceso  
  
'Obteniendo código binario del caracter de cierre cabecera...  
Text2.Text = Text2.Text & Chr(13) & Hour(Now) & ":" & Minute(Now) & ":" & Second(Now) & " -  
Comprovando cabecera de inicio de texto."  
byte2 = 0  
For I = 1 To 8  
Get #1, , byte1 'Obteniendo bytes...  
MisBits(I) = byte1 And &H1  
If Not MisBits(I) = 0 Then  
    'Obteniendo bits del caracter de cabecera...  
    byte2 = byte2 Or bit(I)  
End If  
Next I  
  
'Comprovando si la imagen posee texto...  
If Not byte2 = 178 Then  
    Text2.Text = Text2.Text & Chr(13) & Hour(Now) & ":" & Minute(Now) & ":" & Second(Now) & " - La  
imagen no posee ningun texto."  
    Text2.Text = Text2.Text & Chr(13) & Hour(Now) & ":" & Minute(Now) & ":" & Second(Now) & " - El  
proceso a finalizado inesperadamente.."  
    MsgBox "La imagen introducida no posee texto encriptado.", vbCritical, "TapiporlaPlus"  
    Call borrar  
    Close #1  
    Exit Sub  
End If
```

```
byte2 = 0
Text2.Text = Text2.Text & Chr(13) & Hour(Now) & ":" & Minute(Now) & ":" & Second(Now) & " -
Obteniendo texto de la imagen."
Do While Not byte2 = 178
byte2 = 0
'Obteniendo código binario de cada caracter.
For I = 1 To 8
    Get #1, , byte1 'Obteniendo bytes...
    Progress1.Value = Progress1.Value + 1
    Label3.Caption = "" & Round((Progress1.Value * 100) / Progress1.Max, 1) & " %"
    MisBits(I) = byte1 And &H1
    If Not MisBits(I) = 0 Then
        'Obteniendo bits del caracter...
        byte2 = byte2 Or bit(I)
    End If
Next I
DoEvents
If Not byte2 = 178 Then
'Si el caracter no es el 178, se introduce el caracter en el cuadro de texto.
Text1.Text = Text1.Text & Chr(byte2)
End If
Loop
Close #1
sonido.Command = "play" 'Realizando reproduccion de sonido de finalizado
Text2.Text = Text2.Text & Chr(13) & Hour(Now) & ":" & Minute(Now) & ":" & Second(Now) & " - El
proceso ha finalizado correctamente."
Progress1.Value = Progress1.Max
Label3.Caption = "" & Round((Progress1.Value * 100) / Progress1.Max, 1) & " %"
MsgBox "El proceso ha finalizado correctamente", vbInformation, "TapiporlaPlus"
Set fso = CreateObject("Scripting.FileSystemObject")
msg = MsgBox("¿Desea guardar el texto en un archivo?", 4, "Guardar")
If msg = 6 Then
'En caso de que quiera guardar el archivo en un txt
Dim im
abrir.FileName = ""
abrir.Filter = "TXT (*.txt)|*.TXT"
abrir.ShowSave 'Mostrando ventana de guardar
With abrir
im = .FileName
End With
Set a = fso.CreateTextFile(im, True) 'Creando archivo txt
a.write ("" & Text1.Text) 'Introduciendo texto en el txt
a.Close 'Cerrando archivo
End If
Image9.Visible = True
Image12.Visible = True
End Sub

Private Sub borrar()
'Restableciendo sonido a reproducir
sonido.DeviceType = WaveAudio
sonido.FileName = App.Path & "\Success.wav"
sonido.Command = "Open"
sonido.Command = "Prev"
sonido.Command = "Stop"

'Restableciendo objetos del formulario
foto.Visible = False
Text1.Visible = False
```

```
Text2.Visible = False
Progress1.Visible = False
Label1.Visible = False
Text1.Height = 3135
Image7.Width = 501
Image7.Height = 253
foto.Left = 420
foto.Top = 120
foto.Height = 3855
foto.Width = 7575
agregar.Top = 6840
Text1.Text = ""
Text2.Text = ""
Text3.Text = ""
Label3.Caption = ""
Set Image7.Picture = LoadPicture("")
Image1.Visible = True
Image2.Visible = True
Image3.Visible = True
Image12.Visible = False
End Sub

Private Sub Text1_KeyUp(KeyCode As Integer, Shift As Integer)
'Cambiar contador al soltar una tecla pulsada
Text4.Text = Text1.MaxLength - Len(Text1.Text)
End Sub
```